University of Alberta

APPLICATION OF THE HODGE HELMHOLTZ DECOMPOSITION TO VIDEO AND IMAGE PROCESSING

by

Biswaroop Palit ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Electrical and Computer Engineering

Edmonton, Alberta
Fall 2005

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-494-09255-6
Our file Notre référence
ISBN: 0-494-09255-6

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

To my parents for their unconditional support, love and encouragement.

Abstract

The computation of the 2-D motion field from a sequence of images is one of the key tasks of many vision systems. Analysis and interpretation of flow fields is in general a complex task. One of the most interesting problems is to locate critical points in the motion field. The main theme of this thesis is the identification of critical points in motion fields, which have a physical significance for the corresponding application. The discrete Hodge Helmholtz decomposition is a vector decomposition algorithm which is used in this thesis for locating critical points in a motion field.

Automatic processing of meteorological satellite data is a major field of research. In this context we propose a robust method for automatically detecting and tracking a hurricane eye over a sequence of satellite images. The hurricane eye is a rotational center and is detected as a critical point in the motion field extracted from a hurricane image sequence.

The ridge and valley structure in a fingerprint image can be represented as a vector field. In this thesis, we develop a technique to detect the point of maximum curvature in the fingerprint ridge structure. This is done by analyzing the corresponding vector field and identifying the critical point that corresponds to the maximum curvature point. This point serves as a reference point for the registration of fingerprint images.

Cardiac optical mapping is a relatively new technique for imaging the electrical activity on cardiac tissue. We present different techniques for the analysis of these optically mapped videos. Wavefront morphology and its relation to propagation velocity is studied and physical parameters of the cardiac tissue are predicted. We also compare two different techniques for velocity vector analysis.

Acknowledgements

I would like to thank Dr Anup Basu and Dr Mrinal Mandal for their support and guidance during the course of my studies. I could not have expected more freedom in my work. The credit for introducing me to the field of computer vision, for showing the directions to be explored and providing critical comments on my work goes to Dr Basu. Thanks are due to Dr Mandal for providing the idea to work on the DHHD algorithm, for the innumerable meetings and constant encouragement and for suggesting vital structural changes to this thesis.

Special thanks to Dr Jacques Beaumont, Dr Gil Bub, Dr Matthew Kay and Dr Dana Brooks for replying to my queries over email.

I would take this opportunity to thank my friends and colleagues for their help and support during the last 2 years.

Finally thanks to my family - Baba, Ma, Bonti and Blackie - for their love and for being a constant source of inspiration. Without them this thesis would have been impossible. You may be far away, but I feel your support whenever I need it.

Table of Contents

1 Introduction **1**

1.1 Hurricane Eye Tracking 2

1.1.1 Fingerprint Matching 3

1.1.2 Cardiac Optical Mapping Analysis 4

1.2 Major Contributions 5

1.3 Thesis Organisation 6

2 Review of Related Work: I **7**

2.1 Dense Motion Field Estimation for Fluid Flows 8

2.1.1 Block Matching Algorithm 9

2.1.2 Optic Flow Motion Estimation 9

2.1.3 Fluid Motion Estimation Techniques 11

2.1.4 Structure and Nonrigid Motion Tracking from 2D Images . 13

2.2 Discrete Hodge Helmholtz Decomposition (DHHD) 14

2.2.1 Critical Point Detection Techniques 14

2.2.2 DHHD 16

2.3 Contour Tracking Using Level Set Method 21

2.3.1 Traditional Techniques for Tracking Interfaces 21

2.3.2 Formulation of Level Sets to Represent an Interface 22

2.3.3 Some Advantages of the Level Set Formulation 24

2.3.4 Approximating the Level Set Equation 25

2.3.5 Efficient Front Propagation Schemes 26
2.3.6 Application of Level Sets to Shape Detection 28
2.4 Canny Edge Detection 30
2.5 Fingerprint Analysis 31
2.5.1 Reference Point Detection 33
2.6 Summary 35

3 Review of Related Work: II **36**
3.1 Cardiac Optical Mapping 37
3.1.1 Optical Mapping for Understanding Cardiac Arrhythmias . 40
3.1.2 Application of Image Processing 41
3.2 Analysis of Cardiac Activation Maps 43
3.2.1 Velocity Vector Maps 44
3.3 Simulation of Cardiac Tissue 45
3.4 Summary 48

4 Applications of the Discrete Hodge Helmholtz Decomposition to Image and Video Processing **49**
4.1 Cloud Motion Tracking 50
4.2 Proposed Scheme for Hurricane Tracking 51
4.2.1 Initial Processing 52
4.2.2 Segmentation of the Eye 56
4.2.3 Tracking of the Eye 56
4.2.4 Implementation and Experiments 56
4.3 Algorithm for Fingerprint Reference Point Detection 60
4.3.1 Preprocessing 60
4.3.2 Orientation Field Extraction 62
4.3.3 Experimental Results 67

4.4 Summary 71

5 Cardiac Video Analysis **72**

5.1 Cardiac Video Description 72

5.1.1 Preprocessing 74

5.2 Comparison of Techniques for Cardiac Velocity Vector Analysis . . 75

5.3 Wavefront Curvature Measurement 79

5.3.1 Extracting Isopotential Contours 79

5.3.2 Parameterizing the Isopotential Segments 81

5.3.3 Curvature Calculation 85

5.3.4 Calculation of Velocity 86

5.4 Information Extraction from Optical Maps 88

5.5 Summary 89

6 Conclusions and Future Work **90**

Bibliography **93**

A Code for DHHD **104**

List of Tables

4.1 Gradient values around an extrema point. Here -2.2753 is the extrema point. 55

4.2 Properties of the FVC2000 database 67

4.3 Distance between reference point predicted by DHHD and point predicted manually. 70

5.1 Description of videos 73

List of Figures

2.1 Eigenvalues of the matrix A and corresponding phased portraits [1]. 15

2.2 Decomposition of a synthetic field into its curl-free and divergence-free components. 17

2.3 Harmonic remainder from DHHD. 18

2.4 Detection of singular points. 20

2.5 Overlapping of boundaries. Only "Edge" markers correspond to the propagating interface. 22

2.6 The signed distance function defined for a circular curve. 23

2.7 Propagating front embedded as zero level set 24

2.8 The narrow band shown on a grid [2]. 27

2.9 Detection of arterial structure. 28

2.10 Level set segmentation demonstration 29

2.11 A typical fingerprint image. 32

3.1 A cardiac optical mapping system (from [3]) 38

3.2 Sample frames from the optically mapped video sequence. 40

3.3 Effect of wavefront geometry on propagation velocity [4]. 47

4.1 Motion field in a anticlockwise rotating hurricane sequence extracted using the BMA. 53

4.2 The divergence-free potential function with a distinct maximum and corresponding contours. 54

4.3 Level set extraction of the hurricane eye after 15 iterations (left) and after 300 iterations of the fast marching method(right). 58
4.4 Hurricane eye tracking using level set segmentation: Sequence of images from the Hurricane Luis sequence, with the eye segmented. . 59
4.5 Eye tracking over 100 frames 59
4.6 The result of normalization 61
4.7 Orientation field extracted from a noisy fingerprint image (left) and from the same image after denoising (right). 62
4.8 Gradient directions for a ridge 63
4.9 Orientation field superposed on the region of interest in a fingerprint image. 65
4.10 Discontinuity in orientation field for a fingerprint image. 66
4.11 Squared orientation field, discontinuity is removed. 66
4.12 Divergence-free potential function and corresponding contours for a fingerprint image. 67
4.13 Sample fingerprints from the four databases. 68
4.14 Reference point identified in two different images of the same finger. 69
4.15 Faulty detection of reference point, when actual reference point is not included in the fingerprint. 70
4.16 Images of the same finger with and without reference point, and the detected reference points. 70

5.1 Sample frames from different videos 73
5.2 Preprocessing results. 75
5.3 Original motion field extracted from video sequence. 76
5.4 (a) Divergence-free component of original motion field. (b) Magnitude of the curl of the original field, shown as a color map. 77

5.5 (a) The divergence of motion field shown as a color map with the curl-free field obtained using DHHD is superimposed on the color map. (b) The curl-free component of the field. 78
5.6 Preprocessed frames. 80
5.7 A sample frame from optically mapped video and its extracted isopotential. 80
5.8 Rearrangement of contour points. 82
5.9 Demonstration of regions of curve with higher complexity. 83
5.10 Detected isochronal segments in a sequence of frames. Spline approximation of the segmented contour are superposed on the corresponding frame. 84
5.11 Curvature vectors calculated from parametrized isopotential. 85
5.12 Wavefront velocity vs. wavefront curvature. 87
5.13 Intensity vs. Time plot at a single pixel location for a pig heart. . . . 88

List of Abbreviations

BMA	Block Matching Algorithm
DHHD	Discrete Hodge Helmholtz Decomposition
ECG	Electrocardiogram
GOES	Geostationary Operational Environmental Satellite
VF	Ventricular Fibrillation

Chapter 1

Introduction

The 3-D velocity field in a scene can be observed as a 2-D motion field in a sequence of images. This 2-D motion field is the projection of the 3-D velocity field onto the image plane. Optical flow computation is an important area of research in computer vision. It provides information about the motion and structure of objects. Work in computer vision embraces two broad categories: rigid or slightly deformed bodies, and fluids. A considerable amount of research has been done in the area of rigid bodies [5], [6], [7], [8], [9] as well as for fluid motion [10], [11], [12].

The extraction of higher level descriptors from a flow field is a very important task in fluid motion and vector analysis. This problem is essential in meteorology, for instance, to identify and track depressions or convective clouds in satellite images. The knowledge of all these points is thus precious to understand and predict the flows of interest. Critical points also allow a compact representation of the flow which is very useful in both experimental and theoretical fluid mechanics. The interpretation of flow visualization images and the identification of critical points like vortices is an important research area in computer vision, as discussed in [13] and [14]. Another application of motion interpretation is discussed in [15], where an image is segmented into regions of coherent motion.

The major theme of this thesis is the analysis of vector fields obtained from images or image sequences. We identify critical points in the vector field, which have

a physical significance for the particular set of images. The work in this thesis is divided into three different parts. In this chapter I present the motivation and the contributions of this work along with an overview of the structure of this dissertation.

1.1 Hurricane Eye Tracking

The wide range of remotely sensed data allows one to characterize natural phenomena through different physical measurements. A major advantage of environmental remote sensing is the regular spatial and temporal sampling. Using remotely sensed data, the short range evolution of atmospheric processes can be characterized. However the amount of data received from satellites is huge (approximately one terabyte a day [16]). Computer vision algorithms can be used to organize this huge amount of information and automatically interpret satellite images. This would lead to increasing the reliability of weather surveillance and would facilitate the assimilation of data for numerical weather prediction models.

Spatio-temporal analysis of meteorological images is an important aspect of weather analysis. One of the essential problems is to identify and track hurricanes in satellite images.

A hurricane is like a vortex. It has a distinct central region (called the hurricane eye) around which the remainder of the cloud mass tends to swirl. It is known that the wind speed near the eye of the hurricane has a minima when compared to surrounding areas. This is in contrast to most of the vortex models. Due to this peculiarity, detection of the hurricane core is a difficult problem. Another complexity in the tracking of hurricane features arises from the fact that clouds are deformable in nature.

Almost all current systems for meteorological image data analysis use feature matching for tracking purposes. As an example, Mukherjee and Acton [17] predict

motion vectors describing the motion of clouds using feature points on the boundary of the cloud segments. They use points of inflection on the contour describing a cloud as the feature points. A correspondence between consecutive frames is performed to obtain motion vectors. However, feature matching in general is not suitable for satellite images as they span a large spatial region. Thus a technique which performs tracking on satellite images without using explicit feature matching has a definite advantage over other techniques.

In this thesis a hurricane eye tracking method is presented for a sequence of geostationary satellite images. Given a sequence of satellite images of a hurricane, the objective is to locate the position of the eye accurately in all the frames. Our technique can be applied to the prediction and tracking of the path of a hurricane using data from meteorological satellites.

1.1.1 Fingerprint Matching

Among all biometric techniques, fingerprint based identification is the most mature and proven method [18]. One of the key steps prior to matching two fingerprint images is to calculate a pose transformation for the alignment of the two images. A unique reference point is required for the registration of the images. Singular points of the fingerprint ridge structure have been used for this purpose. However not all fingerprint images have a valid singular point as the part containing the singular point might not have been imaged in some cases.

Most of the singular point detection techniques are based on the orientation field representing the ridge structure of the fingerprint. Such methods use local features like the Poincare index [19]. These techniques are very sensitive to the quality of the fingerprint image and a small amount of noise in the orientation field can lead to erroneous singular points being detected.

Jain *et al.* [20] used the point with maximum curvature on the ridge structure

as a reference point. A multiresolution analysis is proposed in [20] as a robust method to locate reference points. However, this method is sensitive to fingerprint rotation. Moreover, the curvature computed from the orientation field would depend significantly on the noise content in the field.

We propose a technique for fingerprint reference point detection using the discrete Hodge Helmholtz decomposition (DHHD) algorithm. This procedure is robust to noise in the orientation field and can locate the point of maximum curvature irrespective of the fingerprint orientation.

1.1.2 Cardiac Optical Mapping Analysis

Ventricular fibrillation (VF), also known as cardiac arrest, is a highly irregular arrhythmia originating in the ventricles (the lower chambers of the heart). VF causes nearly 250,000 deaths annually in the United States alone [21].

However the mechanism of VF, and the characteristic electrical phenomena observed during VF are not fully understood and is an extensive area of research in the cardiac electrophysiology community. Optical mapping is a relatively new technique developed to study the behavior of electrical impulses in the cardiac tissue. An extracted animal heart is stained with a voltage sensitive dye. This causes the heart to emit fluorescence when electrical waves propagate on the cardiac tissue. Using the optical mapping method, direct observations of the cardiac electrical activity can be made at high spatio-temporal resolution. The analysis of these cardiac electrical patterns is generally done visually. Quantitative analysis of optical mapping data has not been a major area of research until recent years.

A parallel stream of research is the understanding and modelling of the ionic basis of the electrical phenomena observed in the cardiac tissue. Ionic models to simulate the cardiac tissue have been developed [22]. These provide a system of equations governing the propagation of electrical impulses in the heart muscles.

Numerical methods (e.g. finite element based techniques) are used to solve the system of equations proposed by the models.

Based on studies of these electrical models, the existence of spiral wavefronts on the cardiac tissue during VF has been demonstrated [23]. Wavefront curvature is an important parameter of wave propagation. The curvature of a wavefront may cause slowing of the propagation and conduction block. Wavefront curvature is also important for defining the properties of spiral waves which are responsible for some types of cardiac arrhythmias.

Various aspects of electrical wave propagation in cardiac tissue have been studied using simulated models. Kay and Gray [24] developed a technique for measuring the curvature of a wavefront from a data frame of electric potentials obtained by electric models. In [25], velocity vectors for the propagation of electrical impulses are estimated using data from simulation models. The velocity vector field is used to study the conduction properties of the cardiac tissue during fibrillation.

The quantitative analysis of the cardiac videos is still a relatively new field, and most of the existing techniques are still manual. Their analysis is dependent on the pattern recognition skills of the cardiologist. We believe that the quantitative analysis of these videos will lead to significant improvement in understanding the phenomena of fibrillation and promote the analysis of these videos using techniques from computer vision. The objective of our work is to use techniques developed by cardiologists for synthetically generated data, and then to combine them with image processing techniques so that they can be used for the analysis of actual optical mapping data.

1.2 Major Contributions

This thesis focuses on the detection of critical points of a motion field for two specific problems, where the critical point has a physical significance. We also

develop algorithms for the analysis of cardiac optically-mapped videos.

- We develop a framework to localize and track the vortex (eye) of the hurricane. This technique does not require feature matching. Another advantage is that the segmentation of the eye does not require manual initializations.

- A technique for robust fingerprint reference point detection is developed. The new technique for reference point detection is relatively insensitive to noise in the orientation field and is capable of locating a reference point irrespective of the rotation or scaling of the image.

- A technique to detect and parametrically represent the isopotentials on the optically-mapped cardiac video is developed and a curvature velocity analysis for electrical waves on the cardiac tissue is performed.

1.3 Thesis Organisation

The remainder of the thesis is organized as follows. Review of related literature for hurricane eye tracking and for the fingerprint reference point location are provided in Chapter 2. In Chapter 3 we discuss cardiac optical mapping and some techniques used for analyzing cardiac electrical propagation. The proposed algorithm for application of DHHD to hurricane eye tracking and to fingerprint reference point location is discussed in Chapter 4. The framework for quantitative analysis of cardiac videos is presented in Chapter 5. Chapter 6 contains the conclusions of this thesis, followed by some suggestions for future work.

Chapter 2

Review of Related Work: I

The analysis of video sequences showing the evolution of fluid phenomena has attracted a great deal of attention from the computer vision community. The applications concern domains such as experimental visualization in fluid mechanics, environmental sciences (oceanography, meteorology, *etc.*), or medical imagery. Analyzing fluid motion is essential in a number of domains and can rarely be handled using generic computer vision techniques. We first provide a review of motion estimation techniques, focusing on the estimation of dense velocity maps from image sequences.

However, just the velocity field is far from being the ultimate goal of an analysis. Differential or integrated information from the velocity field is more valuable for concerned experts [26]. For example, it is essential to characterize fluid flows to extract the vorticity fields, the streamlines, or the singular points of the flows. All these features may be estimated indirectly from the velocity field by differentiation or by integration. The irrotational and the solenoidal components of the velocity fields provide information about the vorticity and the divergence of the fields and the location of the singular points of major interest (namely sources, sinks and vortexes). The Hodge Helmholtz decomposition is a very effective technique for locating such critical points.

We use the Hodge Helmholtz decomposition for the analysis of motion fields

extracted from images and use it to identify critical points in the field. These points have a physical significance in the corresponding application. We use the DHHD algorithm to detect the hurricane eye in a hurricane image sequence. This is segmented out using a level set segmentation algorithm. We also analyse the orientation field representing a fingerprint image using the DHHD method to identify a reference point.

In this chapter, we review the techniques used in Chapter 4 of this thesis. A review of motion detection techniques, especially the techniques developed for detecting fluid motion, is presented in Section 2.1. Section 2.2 gives an overview of the DHHD algorithm and its implementation. We use the DHHD algorithm for the hurricane eye detection and fingerprint reference point detection. Level set technique and its implementation is described in Section 2.3. We use a level set based segmentation method to extract the region corresponding to the eye of the hurricane. Section 2.5 reviews techniques for fingerprint analysis and reference point detection. Section 2.6 summarizes this chapter.

2.1 Dense Motion Field Estimation for Fluid Flows

In the analysis of the hurricane video sequence, as well as while studying the cardiac video sequence (described in Chapter 3), the motion is similar to fluid motion activity. The computation of the apparent image velocity by processing sequences of images is one of the fundamental problems in computer vision. A very good review of various optical flow estimation techniques is presented in [7]. Here, I present a review of motion estimation techniques, focusing on techniques designed especially for fluid motion.

2.1.1 Block Matching Algorithm

The Block Matching Algorithm (BMA) is the most common motion estimation method and is used in most video coding standards (MPEG, H.261/262/263). The basic assumption involved here is that a block of pixels undergoes the same translational motion from one frame to the next. Each frame is partitioned into square blocks of size $m \times m$. Searching boundaries are defined by $\triangle_h$ and $\triangle_v$ and are centered at the corresponding block center. The block is shifted inside the search area and a matching measure is calculated for each position of the block. The position which gives the best value of the matching function is chosen as the position of the block in the next frame. Examples of match measure functions are normalised cross-correlation, absolute mean difference and mean squared difference. We use the following matching function:

$$(\hat{u}, \hat{v}) = \min_{|u| \leq \triangle_h, |v| \leq \triangle_h} \sum_m \sum_n |I_{t+1}(x+u, y+v) - I_t(x, y)| \qquad (2.1)$$

For each block a displacement vector is obtained using Eqn. (2.1). The BMA has been used for cloud detection in [27]. The advantage of BMA over optic flow based techniques (discussed next) is their ability to handle large displacements. Large displacements can be detected using a large search area although this comes at a price of increased computational complexity.

2.1.2 Optic Flow Motion Estimation

The seminal work of Horn and Schunck [5] on optic flow calculation is one of the most cited in the field of motion estimation. Optic flow resorts to the minimization of an objective functional composed of two terms. The optical flow cost function is of the form:

$$C = \int_\Omega f_1[\nabla E(x, y, t) \cdot v(x, y, t) + \frac{\partial E(x, y, t)}{\partial t}] + \alpha f_2[|\nabla v(x, , y, t)|] \qquad (2.2)$$

where v is the unknown velocity, Ω is the image plane, E is the image brightness, f_1 and f_2 are two penalty functions. The first component in Eqn. 2.2 is called the data term. It is based on a brightness constancy assumption, *i.e.* visible points conserve their intensity in the course of a small displacement. This can be represented as:

$$\frac{dE}{dt} = \nabla E \cdot v + \frac{\partial E}{\partial t} \approx 0 \tag{2.3}$$

The penalty function is generally chosen as the least square estimate, *i.e.* minimize $\left(\nabla E(x,t) \cdot v(x,t) + \frac{\partial E(x,t)}{\partial t}\right)^2$. There might be regions in the image where the brightness constancy assumption does not hold, like occlusion boundaries or regions with multiple motions. As discussed in [8], penalty functions obtained from robust statistics can be used which reduce the influence of outliers. These modified penalty functions are used to detect multiple motions in the same scene.

The second term in Eqn. 2.2 is a regularization term which enforces smoothness of the solution. This is required to constrain the system of equations. To gain accurate estimates, this region should be sufficiently large to constrain the solution. However, the larger the region of integration, the more likely it is to contain multiple motions (this is known as the generalized aperture problem). A framework to handle this problem, using robust statistics is proposed in [8] and [28].

The differential nature of the brightness constancy equation makes it unsuitable for use when a large displacement occurs between consecutive frames. To handle large displacements, the brightness conservation is expressed in an integrated way:

$$E(x + d(x), t + \triangle t) - E(x,t) \approx 0 \tag{2.4}$$

where $\triangle t$ is the interval between consecutive frames and $d(x)$ is the displacement.

Due to the great deal of spatio-temporal distortions that intensity patterns exhibit in images of fluids and the compressible nature of fluids, standard methods like optic flow are not well adapted for these problems. A second contradictory scenario

occurs because of motion which is not parallel to the visualization plane, *i.e.* fluid elements entering or exiting the image slice. Another problem might arise due to oversmoothing. Divergence and vorticity are known to be large in certain regions of the fluid flow, and the first order regularization might lead to oversmoothing and hence loss of important characteristics of the flow field.

2.1.3 Fluid Motion Estimation Techniques

To overcome the above problems, dense motion field estimators for fluid flows have been studied extensively by the computer-vision community. Recently functionals dedicated to fluid images have been proposed. Corpetti *et al.* propose a scheme for fluid motion extraction from image sequences in [10] and [29]. They incorporate a data-term based on the continuity equation of fluid mechanics instead of the usual brightness constancy assumption in Eqn. 2.2. Additional improvements are obtained by considering tailor-made regularization terms preserving the concentrations of divergence and vorticity.

Image brightness for a fluid is usually related to its density. The density ρ obeys the following continuity equation:

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho V) = 0 \tag{2.5}$$

where V is the three dimensional velocity field. By analogy the brightness continuity equation is:

$$\frac{\partial E}{\partial t} + \text{div}(Ev) = 0 \tag{2.6}$$

An integrated version of the constraint is considered, so that it can be used to estimate large displacements:

$$E(x + d(x), t + \triangle t) = E(x, t)exp(-\text{div}d(x)) \tag{2.7}$$

We see that the brightness is scaled by a factor of $exp(-\text{div}d(x))$. The term $E(x,t)exp(-\text{div}d(x))$ decreases (respectively, increases) for motion with positive

(respectively, negative) divergence. For zero divergence, it equals the brightness constancy constraint.

The assumption of the global conservation of density might be physically violated in some cases such as presence of sinks and sources, 3-D motion nonparallel to the image plane *etc*. Hence, a soft penalty function is used, which will limit the impact of various violations of the continuity equation on the overall motion estimate.

To avoid oversmoothing, a second-order regularization is used. The regularization term is the second term in Eqn. 2.2. The constraint used here is different from the standard smoothness constraint, which is generally used for regularization. The basic form used is:

$$\int_{\Omega} |\nabla div(d)|^2 + |\nabla curl(d)|^2 \tag{2.8}$$

The estimation of the dense displacement field is carried out on a multiresolution framework. At a given resolution, the displacement field obtained from the previous resolution is refined. The approximate solution obtained is added to the estimated motion field and passed to the next resolution.

Another approach to measuring fluid flow from image sequences is presented in [11]. The approach draws upon principles from fluid mechanics for the motion recovery algorithm. The algorithm is constrained by the conservation of mass continuity equation:

$$E_x u + E_y v + E u_x + E v_y + E_t = 0 \tag{2.9}$$

E_x and E_y are the spatial partial derivatives of the image brightness E. u_x and v_y are the partial derivatives of the velocity components.

Such information derived from the physical behavior of fluids is used to motivate a flow-recovery algorithm. The Rankine model of flows is used in [14] to obtain a parametric description of the critical points in motion fields. The Rankine

vortex is mathematically described by:

$$V_r = 0 \tag{2.10}$$

$$\begin{aligned} V_\theta &= \omega r \quad (r \leq r_0) \\ V_\theta &= \omega \frac{r_0^2}{r} \quad (r > r_0) \end{aligned} \tag{2.11}$$

where r = radius from the rotation axis (z-axis), ω = angular velocity of any fluid particle within r_0. V_r is the radial component of the velocity and V_θ is the angular velocity. The velocity field associated to a source/sink in the plane is modelled using a similar scheme.

2.1.4 Structure and Nonrigid Motion Tracking from 2D Images

Cloud motion is a special case of fluid motion and consists of very complex motion dynamics. Zhou *et al.* [30] and [31] use a combination of local and global models to obtain accurate cloud-top structure and motion.

The local motion analysis performs nonrigid motion tracking within a small area and aims to capture small nonrigid details. The image is divided into small blocks and an affine motion model for each block is estimated by evaluating an error of fit function (since the affine motion model takes into account transformations such as rotation, scaling, shearing etc). The assumption is that local motion associated with each region can be approximated by an affine transformation of the corresponding region in the previous frame.

The structure and motion recovered after the local motion analysis may have sharp discontinuities. It also ignores interaction between neighboring areas. Thus the global constrains are essential to limit possible nonrigid behaviors and to regularize the locally recovered parameters. The global shape model provides extra constraints which incorporate prior knowledge about a shape's smoothness and its resistance to deformation. The local and global constrains are used iteratively to obtain an accurate description of the motion field.

The results shown in [30] do match the motion vectors of the actual image. However the prediction in the area of the hurricane eye is not very good. This is because the fluid dynamics in the eye is very complex and cannot be tracked by the affine motion model.

In the analysis of hurricane video, we have tried various flow detection algorithms. However, the performance of our technique is satisfactory with a BMA estimated motion field. Hence we have used the BMA while presenting the results for the hurricane eye tracking method.

2.2 Discrete Hodge Helmholtz Decomposition (DHHD)

Higher level descriptors of a vector field, which convey meaningful information about the field, are an important area of study. This analysis leads to qualitative description of the vector field. In this section, we discuss various techniques that are used to detect singular points in a vector field with emphasis on the DHHD.

2.2.1 Critical Point Detection Techniques

Rao and Jain [13] presented a scheme to locally approximate the flow using 2-D linear differential equations which is then used for the analysis of the flow field. The phase portraits depicting these differential equations represent the trajectory of the particles of the fluid (phase portraits represent the solution curves for the system of equations). Critical points like source/sink and rotational centers are obtained by analyzing the phase portraits. A non-linear least squares technique was used to estimate a first-order flow model from the oriented flow field.

A similar analysis was developed by Ford and Strickland [32]. Here the motion field in a local window is represented by an affine model:

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \mathbf{A}\vec{X} + \mathbf{b} \tag{2.12}$$

Here $\vec{X} = (x, y)$ is the pixel location, $u(x, y)$ and $v(x, y)$ are the velocity components at the corresponding pixel locations and A is the 2×2 affine matrix. The elements of A vary for each pixel location. Critical points can be determined according to the eigenvalues of matrix A. The relationship between the phase portrait structure and eigenvalues of A is presented in Fig. 2.1.

Eigenvalues of A	Jordan form	Type	Phase portrait
$\lambda_1 \neq \lambda_2$, and both of them are real number	$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$, and λ_1 and λ_2 have same sign	Node	
	$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$, and λ_1 and λ_2 have opposite sign	Saddle	
$\lambda_1 = \lambda_2$	$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_1 \end{bmatrix}$	Star-node	
	$\begin{bmatrix} \lambda_1 & 1 \\ 0 & \lambda_1 \end{bmatrix}$	Improper node	
$\lambda_{1,2} = \alpha \pm i\beta$	$\begin{bmatrix} 0 & -\beta \\ \beta & 0 \end{bmatrix}$, $\alpha = 0$	Center	
	$\begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix}$, $\alpha \neq 0$	Spiral	

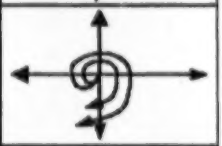



Figure 2.1: Eigenvalues of the matrix A and corresponding phased portraits [1].

Cohen and Herlin [33] presented a method for approximating an orientation field and characterizing the stationary points of the trajectories using an arbitrary polynomial phase portrait. The Poincare index (Eqn. 2.35) is used to locate and characterize the critical points. Nogawa *et al.* [12] proposed a technique for the

analytic modelling of the flow field and used the Cauchy theorem of residues to detect vortices. The Cauchy residue theorem evaluates an integral along a closed curve. This integral is independent of its path, and determined only by the residues of singular points inside the curve. After the critical points have been identified, a Rankine vortex model is used to obtain a parametric representation of the field.

The linear model described by Rao and Jain [13] can handle only one critical point. The nonlinear phase portrait model, used in [32] and [33], can handle multiple critical points, but these models are computationally expensive. The approach of Nogawa *et al.* [12] is very sensitive to noise. These inherent problems have led to exploration of different techniques for the analysis of vector fields.

2.2.2 DHHD

The Hodge-Helmholtz decomposition is a technique used to decompose a continuous vector field into a curl-free component, a divergence-free component and a harmonic remainder. The technique has been extended to discrete vector fields in [34] and to 3D discrete fields in [35]. A field $\vec{\xi}$ is decomposed as:

$$\vec{\xi} = \vec{\xi}_{CF} + \vec{\xi}_{DF} + \vec{\xi}_{HR} \tag{2.13}$$

where $\vec{\xi}_{CF}$ is the curl-free component, $\vec{\xi}_{DF}$ is the divergence-free component and $\vec{\xi}_{HR}$ is the harmonic remainder. Associated with the first two components, we have potential functions E and $\vec{W}$:

$$\vec{\xi}_{CF} = \nabla E; \quad \vec{\xi}_{DF} = \nabla \times \vec{W} \tag{2.14}$$

Here E is a scalar potential and $\vec{W}$ a vector potential. The harmonic remainder satisfies $\nabla \cdot \vec{\xi}_{HR} = 0$ and $\nabla \times \vec{\xi}_{HR} = 0$. ∇, $\nabla\cdot$ and $\nabla\times$ represent the gradient, the divergence and the curl operators, respectively. For a function ϕ, the gradient, divergence and curl operators are defined by:

$$\nabla\phi(x,y,z) = \frac{\partial\phi}{\partial x}\hat{x} + \frac{\partial\phi}{\partial y}\hat{y} + \frac{\partial\phi}{\partial z}\hat{z} \tag{2.15}$$

$$\nabla \cdot \phi(x,y) = \frac{\partial\phi_x}{\partial x} + \frac{\partial\phi_y}{\partial y} + \frac{\partial\phi_z}{\partial z} \tag{2.16}$$

$$\nabla \times \phi = \begin{vmatrix} \hat{x} & \hat{y} & \hat{z} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ \phi_x & \phi_y & \phi_z \end{vmatrix} \tag{2.17}$$

Here ϕ_x, ϕ_y and ϕ_z are the x, y and z components of the function, ϕ. $\hat{x}$, $\hat{y}$ and $\hat{z}$ represent unit vectors along the coordinate axis.

A method for dense motion field estimation in fluid images using the Helmholtz decomposition is described in [26]. The goal of the approach is to perform a direct approximation of the two potential functions E and $\vec{W}$ (Eqn. 2.14) from the brightness constancy constraint equation. This is in contrast to the normal procedure of calculating the flow field first used in the techniques in Section 2.1.3.

Fig. 2.2 is an example of how DHHD works. A synthetic field is decomposed into its curl-free and divergence-free components. The corresponding harmonic remainder is shown in Fig. 2.3. The magnitude of the harmonic field is 0.59% of the original vector field. The procedure to analyze the decomposed field and its corresponding potential functions in order to predict singularities in the motion field is described next.

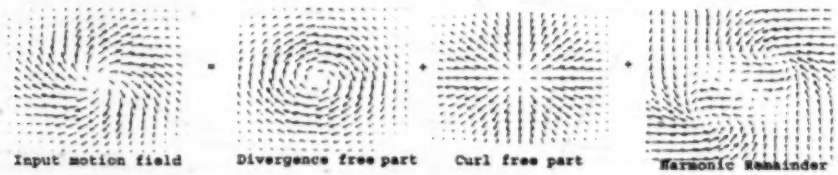


Figure 2.2: Decomposition of a synthetic field into its curl-free and divergence-free components.

Previous versions of DHHD in [34] and [35] have been implemented using irregular grids for the analysis of the discrete vector fields. Guo *et al.* demonstrated in [36] that the use of regular grids greatly simplifies the computational complexity

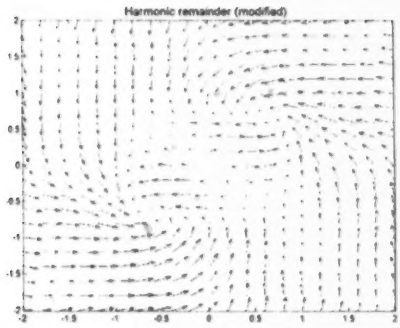


Figure 2.3: Harmonic remainder from DHHD.

of the algorithm and preserves sufficient accuracy. A regular triangular grid is used in their implementation of DHHD for the analysis of motion fields.

The input to DHHD is a vector field $\vec{\xi}$ with $M \times N$ vectors. An $M \times N$ grid is defined, with $(M-1)\times(N-1)$ rectangles, such that each vertex has a corresponding motion vector. Each of these rectangular blocks is divided into two triangular blocks to obtain a regular triangular grid. The velocity vectors have to be redefined for the new grid. Each triangular block is assigned a motion vector equal to the average of the motion vector at its three vertices. The new vectors are labelled $\vec{\xi}_k$, where $k \in [1, 2, \ldots, 2 \times (M-1) \times (N-1)]$. The curl-free and the divergence-free potential functions, of (2.14), at the nodes of this triangular mesh are represented by:

$$E = \begin{bmatrix} E_1 & E_{M+1} & \cdots & E_{(N-1)*M+1} \\ E_2 & E_{M+2} & \cdots & E_{(N-1)*M+2} \\ \vdots & \vdots & \vdots & \vdots \\ E_M & E_{2M} & \cdots & E_{N*M} \end{bmatrix} \tag{2.18}$$

$$W = \begin{bmatrix} W_1 & W_{M+1} & \cdots & W_{(N-1)*M+1} \\ W_2 & W_{M+2} & \cdots & W_{(N-1)*M+2} \\ \vdots & \vdots & \vdots & \vdots \\ W_M & W_{2M} & \cdots & W_{N*M} \end{bmatrix} \tag{2.19}$$

Concatenating the matrices columnwise, we get vectors $E' = (E_1, E_2, \cdots, E_{M*N})^T$

and $W' = (W_1, W_2, \cdots, W_{M*N})^T$. The curl-free component of the field $\vec{\xi}_{CF}$ is the projection of the original field $\vec{\xi}$ onto the space of curl-free fields. It can be obtained by minimizing the distance between $\vec{\xi}$ and ∇E over the entire image domain (Ω). A similar algorithm is used to calculate the divergence-free field $\nabla \times \vec{W}$. The distance can be expressed by:

$$D(E) = \int_\Omega ||\nabla E - \vec{\xi}||^2 d\Omega, \quad D(\vec{W}) = \int_\Omega ||\nabla \times \vec{W} - \vec{\xi}||^2 d\Omega \tag{2.20}$$

The minimum distances, $D(E)$ and $D(\vec{W})$ are found by considering the partial derivatives $\frac{\partial D(E)}{\partial E_i}$ and $\frac{\partial D(\vec{W})}{\partial \vec{W}_i}$. Finite element method (FEM) analysis is used to solve for the elements of E' and W', as described in [36]. The minimization of the distance functions in Eqn. 2.20 is simplified to:

$$\int_\Omega \nabla \phi_i ||\nabla E - \vec{\xi}|| d\Omega = 0 \tag{2.21}$$

This simplifies over the finite grid to:

$$\sum_{T_k} \nabla \phi_{ik} \cdot (\nabla E)_k = \sum_{T_k} \nabla \phi_{ik} \cdot \vec{\xi}_k \tag{2.22}$$

where T_k is a triangular element on the image domain, and $\nabla \phi_{ik} = (y_b - y_a, x_b - x_a)$ and (x_a, y_a) are the coordinates of the grid points. Expanding both sides of Eqn. 2.22 we obtain:

$$A_1 E' = B, \quad A_2 W' = C \tag{2.23}$$

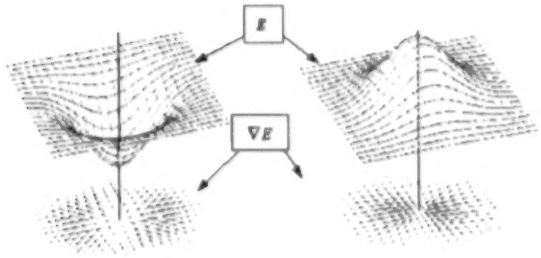
A_1 and A_2 are $L \times L$ sparse matrices ($L = M * N$), which are independent of the input vector field (but depend on basis functions defined over the grid in the FEM analysis, *i.e.* depends on $\nabla \phi_i$). B and C are $L \times 1$ vectors determined by the motion field $\vec{\xi}$ and can be calculated from the right side of Eqn. 2.22.

Reorganizing back to the matrix form, we obtain E and W. Now $\nabla \times \vec{W}$ and ∇E can be calculated using simple vector operations:

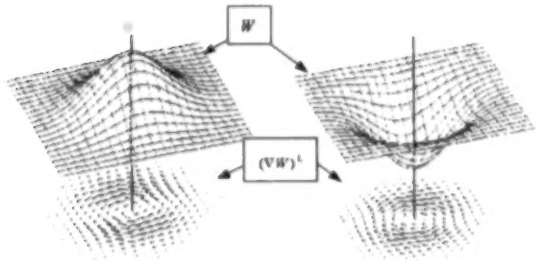
$$\nabla E \quad \text{and} \quad (\nabla W)^\perp = [(\nabla W)_y, -(\nabla W)_x] \tag{2.24}$$

Finally, the harmonic remainder is computed as:

$$\vec{R} = \vec{\xi} - \nabla E - (\nabla \times \vec{W}) \tag{2.25}$$



(a) Detection of source and sink



(b) Detection of rotational center

Figure 2.4: Detection of singular points.

Critical points like sources, sinks and rotational centers can be characterized on the basis of certain properties of the potential functions. A critical point is a point where the velocity of the flow is zero. At an extremum point of the potential function E, we have $\frac{\partial E}{\partial x} = 0$ and $\frac{\partial E}{\partial y} = 0$. Thus, ∇E is zero at this point, or the curl-free component of the field is zero. Hence, an extremum in the curl-free potential surface corresponds to a critical point in the curl-free field. The criteria for detecting a critical point are:

- Point $\vec{p}$ is a source (velocity field diverges outwards from this point) if $E(\vec{p})$ is a minimum

- Point $\vec{p}$ is a sink (field vectors converge to this point) if $E(\vec{p})$ is a maximum.
- A point $\vec{p}$ is an anticlockwise rotational center if $W(\vec{p})$ is a maximum
- Point $\vec{p}$ is a clockwise rotational center if $W(\vec{p})$ is a minimum

Some synthetic fields and their corresponding potential functions are shown in Fig. (2.4). This demonstrates how the potential functions can be used to identify critical points in a field.

2.3 Contour Tracking Using Level Set Method

Image segmentation and object extraction are among the most well addressed topics in computational vision. Curve propagation techniques have been used to address the task of image segmentation. The central idea behind such an approach is perform image partition through planar curves/surfaces.

In this section, I present an overview of the level set method and its application to image segmentation and modeling. The level set method is a numerical technique for tracking interfaces and shapes. Level set method falls into the category of active contour techniques like snakes [37] since the contour can be given any desired shape by applying a halting criteria synthesized from the image data.

2.3.1 Traditional Techniques for Tracking Interfaces

The basic problem in curve propagation is to track the motion of an interface (a dynamic boundary) as it moves in a direction normal to itself with a given speed F.

One of the standard techniques for curve tracking is the marker technique. A number of markers are placed at close intervals on the curve. Each marker at time instant (or iteration number) n represents the point (x_i^n, y_i^n) on the moving front. A numerical algorithm that can estimate (x_i^{n+1}, y_i^{n+1}) from the previous position is used to track the front. The position of the markers at any instant of time is

then used to reconstruct the front. Snakes [37] have a similar formulations as the boundary or contour is represented in a parametric form.

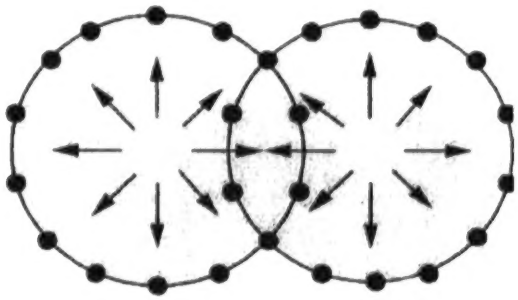


Figure 2.5: Overlapping of boundaries. Only "Edge" markers correspond to the propagating interface.

The marker based technique is a fast and effective scheme to track the motion of curves for simple motion. However, this technique runs into problems when tracking complex motion patterns, *e.g.* motion which involves the overlapping of boundaries (a topological change in the moving front). It is very difficult to keep track of which markers still belong to the boundary and which are inside. This problem can be attributed to the fact that the marker method uses a local representation of the curve instead of a global one.

2.3.2 Formulation of Level Sets to Represent an Interface

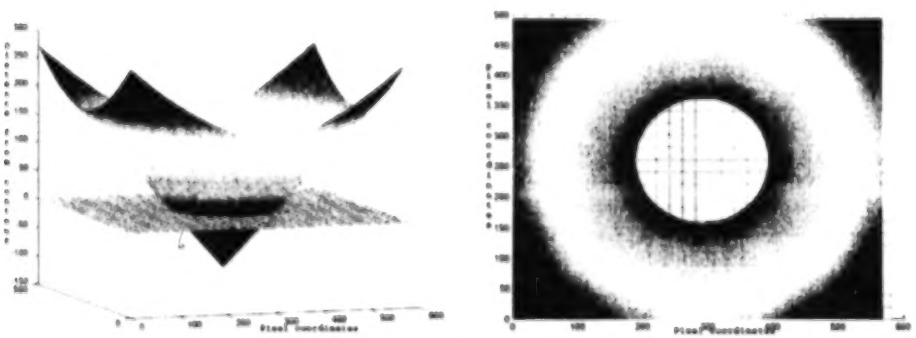
Level set methods fall into a different category of tracking techniques, where the properties of the contour are defined on a fixed cartesian grid. The level set method tries to overcome the problems that traditional tracking methods run into. The basic idea of level sets is to represent the propagating interface as the zero level set of some higher dimensional function ϕ (*e.g.* for a 2D problem we evolve a 3D function to solve the tracking problem). Let $x(t)$ be a point on the curve $\gamma(t)$ at time t. The constraint that has to be satisfied is:

$$\phi(x(t), t) = 0 \tag{2.26}$$

This constraint implies that the set of points where $\phi(x,t) = 0$ represents the current position of the curve). The set of points which satisfy $\phi(x,t) = 0$ are called the zero level set of the function $\phi(x(t))$. ϕ is initialized as:

$$\phi(x, t = 0) = \pm d \tag{2.27}$$

where d is the distance of point x from the curve $\gamma(t = 0)$ (the initial curve). The convention for the sign is that the distance is negative (positive) if point x is inside (outside) the initial curve. A sample distance function for a circular curve is shown in Fig. 2.6. It shows a plot of the signed distance function over all image pixels. Note that points within the circular curve are negative while those outside are positive.



(a) Distance function from side (b) View from top

Figure 2.6: The signed distance function defined for a circular curve.

Using the chain rule on the constraint Eqn. (Eqn. 2.26), we have:

$$\phi_t + \nabla\phi(x(t), t) \cdot x'(t) = 0 \tag{2.28}$$

ϕ_t is the partial derivative of the distance function. The direction normal to the front is represented by $\vec{n}$. The normal is defined as $\vec{n} = \nabla\phi/|\nabla\phi|$. Since $x'(t)$ is the velocity of point x, $x'(t) \cdot \vec{n} = F$ is the speed in the outward normal direction.

Substituting these we get the evolution equation for the curve as:

$$\phi_t + F|\nabla\phi| = 0 \tag{2.29}$$

For a two dimensional curve the above initial value partial differential equation is equivalent to:

$$\phi_t + F(\phi_x^2 + \phi_y^2)^{1/2} = 0$$

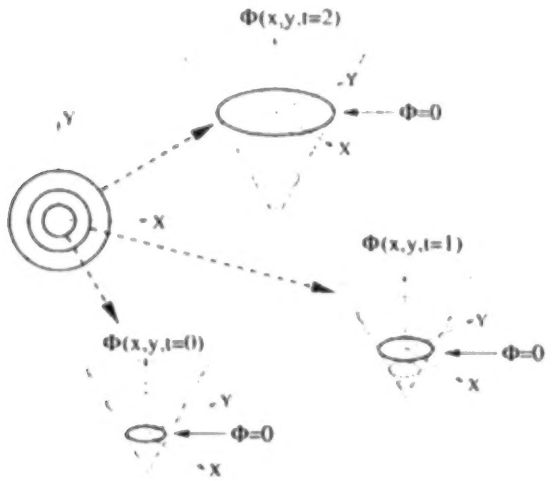


Figure 2.7: Propagating front embedded as zero level set

The value of the level set function is stored in each grid point. As the curve moves, the values are updated. Fig. 2.7 shows a circular curve expanding outwards. At time instant $t = 0$, it is a small circle and the corresponding level set is marked on the signed distance function. As the curve evolves, the zero level set moves upward on the cone representing the signed distance function.

2.3.3 Some Advantages of the Level Set Formulation

Intuitively, the conversion of a curve tracking problem into a surface tracking problem seems to be adding unnecessary complexity to a simple problem. However, in this case the addition of an extra dimension simplifies the analysis to a great extent.

The evolution Eqn. (Eqn. 2.29) ensures that the function $\phi(x,t)$ remains a smooth function and does not develop any singular points. The evolution equation ensure that as long as the the speed F is smooth, $\phi(x,t)$ remains smooth as well. However the curve $\gamma(t)$ (or equivalently the level surface $\phi = 0$) can break, merge or form sharp corners.

Level sets also facilitate the easy calculation of the geometrical properties of the interface. For example, the expression for the curvature of the front is:

$$\kappa = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|} = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^3/2} \tag{2.30}$$

The final shape recovered by the snake method [37] depends significantly on the initial seeding points. This is because the energy functionals used in for the snake evolution are non-convex and hence have multiple local minima. Therefore, an initial guess which is reasonably close to the desired shape is essential for convergence to a satisfactory solution. For level set methods, the convergence to the final shape is relatively independent of the shape initialization as described by Malladi *et al.* [38].

2.3.4 Approximating the Level Set Equation

The mathematical basis for solving the level set equations are drawn from the hyperbolic conservation laws. The overall speed function can be decomposed as $F = F_{prop} + F_{curv} + F_{adv}$ where $F_{prop} = F_0$ is the speed with which the curve expands in the normal direction, $F_{curv} = -\epsilon\kappa$ is the curvature dependent speed and $F_{adv} = \vec{U}(x,y,t) \cdot \vec{n}$ is the speed with which the front is being passively advected. The advection component tends to shift the entire contour along a particular direction while the other two velocity terms determine the shape of the evolving contour. Note that $\vec{U}(x,y,t)$ is a time and position dependent velocity field and $\vec{n}$ is the normal to the front. Substituting these values in the speed function of Eqn. 2.29, the

level set evolution equation can be rewritten as:

$$\phi_t + F_0|\nabla\phi| + \vec{U}(x,y,t)\cdot\nabla\phi = \epsilon\kappa|\nabla\phi| \tag{2.31}$$

The simplest approach is to use the central difference operator to approximate each of the derivatives. The following equation is a complete first-order convex scheme to approximate the level set evolution ([39]):

$$\begin{aligned}\phi_{ij}^{n+1} = \phi_{ij}^{n} + \Delta t\Big(&max(D_{ij}^{-x}\phi,0)^2 + min(D_{ij}^{+x}\phi,0)^2 \\ &+max(D_{ij}^{-y}\phi,0)^2 + min(D_{ij}^{+y}\phi,0)^2\Big)^{1/2}\end{aligned} \tag{2.32}$$

Here $K_{i,j}^n$ is an approximation to the curvature expression in Eqn. 2.30 (using central differences). The difference operatiors are given by:

$$\begin{aligned} D_{ij}^{+x}\phi &= \frac{\phi_{i+1,j}-\phi_{i,j}}{\Delta x} \\ D_{ij}^{-x}\phi &= \frac{\phi_{i,j}-\phi_{i-1,j}}{\Delta x} \end{aligned}$$

This scheme takes into account the entropy conditions, which are necessary to handle the cases when the contour forms sharp corners or overlaps.

2.3.5 Efficient Front Propagation Schemes

The level set evolution Eqn. (2.33) involves $O(N^3)$ calculations (where N is the number of points on the grid). The **narrow band** and **fast marching** approaches are aimed to reduce the computational complexity of the level set evolution.

Since the vicinity of the zero level set is the main region of interest in the evolution of ϕ, the signed distance function is calculated only in the domain of a **narrow band** around the current position of the front. Outside the domain, the distance is defined as $\pm maxDist$. $maxDist$ is the distance of the outermost point on the narrow band from the front. The computational advantage gained over here is because

the points outside the narrow band (*i.e.* points which do not influence the shape of the curve) are not updated. The narrow band structure is shown in Fig. 2.8. The dark dots are points within the narrow band while the hollow dots are points outside the narrow band which are all considered to be $maxDist$ away from the front.

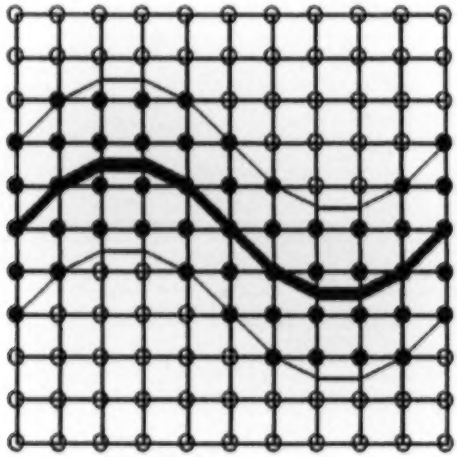


Figure 2.8: The narrow band shown on a grid [2].

The level set Eqn. 2.33 is updated only at points within the narrow band. The narrow band is re-initialized as the interface nears the edge of the band. Re-initialization is such that the zero level set boundary is at the center of the new band. Now $O(kN^2)$ computations are required for tracking the interface, where k is the number of cells in the narrow band grid defined.

The fast marching method is applied when the speed F is either always positive or always negative. This implies that information always flows only in one direction (from smaller values of T to larger values in the case where F is positive). If a solution is built outward from the smallest value of T, an iterative implementation is not required and the solution is constructed in a single pass. The fast marching and narrow band schemes can be combined to yield a computationally efficient evolution technique with complexity of order $O(N^2)$.

2.3.6 Application of Level Sets to Shape Detection

Level set methods have been applied to problems in image processing with applications in image segmentation and shape modelling [38]. The basic approach is to define initial fronts inside the region of interest. These are then evolved using the level set approach. The speed function is determined by the image characteristics. A characteristic that is required for such a speed function is that it must stop at the object boundary. This is achieved by using information about the image gradient. The speed function is multiplied by the term k_I, which serves as a stopping criteria:

$$k_I(x,y) = \frac{1}{1+|\nabla G_\sigma * I(x,y)|} \tag{2.33}$$

Figure 2.9: Detection of arterial structure.

Here G_σ represents a Gaussian filter with standard deviation σ. The term $\nabla G_\sigma * I(x,y)$ is usually close to zero but when the image gradient changes rapidly it has a large value. Therefore $k_I(x,y)$ is approximately unity except near the sharp boundaries where it has a small value.

If we multiply the speed function with the filter $k_I(x,y)$, we are essentially retarding the front at the boundaries. The modified equation of evolution is given by:

$$\phi_t + K_I \cdot F|\nabla\phi| = 0 \tag{2.34}$$

Thus, assuming that the shape we are trying to detect has a distinct boundary,

this method provides an efficient means to stop the evolving curve when the boundary is reached. The advantage here is that no previous information about the shape boundary is required for the segmentation. The initialization just consists of identifying a point within the region of interest. An example of segmentation using level set evolution with image gradient dependent speed function is shown in Fig. 2.9.

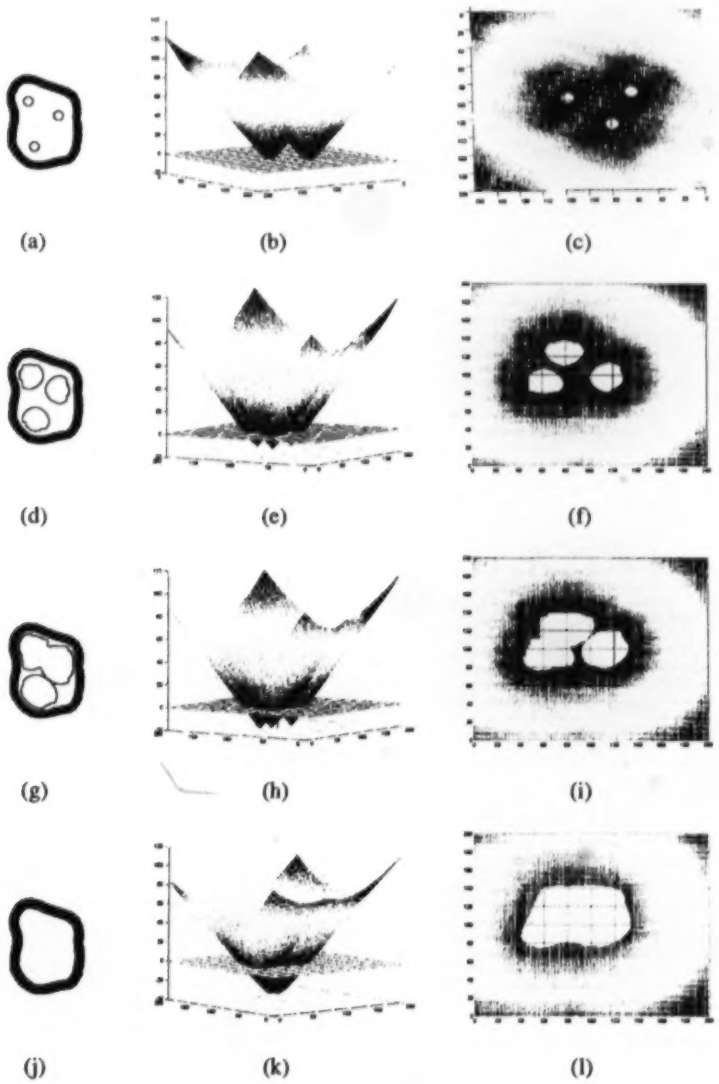


Figure 2.10: Level set segmentation demonstration

Fig. 2.10 depicts a level set segmentation of a ring like structure. Figs. 2.10(a), 2.10(d), 2.10(g), 2.10(j) in the leftmost column represent the segmented area after 200, 1800, 3500 and 5000 iterations, respectively, of the fast marching method. The corresponding signed distance functions are shown in Figs. 2.10(b), 2.10(e), 2.10(h), 2.10(k). The plane in the middle is the $x - y$ plane. Points of intersection with this plane correspond to the zero level set of the surface. Points inside the curve have a negative distance function and are below the zero plane. Figs. 2.10(c), 2.10(f), 2.10(i), 2.10(l) show the same distance function viewed from top. The clear area in these images is the $x - y$ plane visible because the signed distance function is below the plane at these points. The shaded region is the signed distance function covering the $x - y$ plane. The segmenting curve is shown to be the same as the intersection of the signed distance function with the $x - y$ plane.

In this thesis, we use the level set segmentation method for segmenting the eye of the hurricane. The initial seed point for this algorithm is provided by the DHHD algorithm. These are described in further detail in Chapter 4.

2.4 Canny Edge Detection

The Canny operator is a commonly used edge detector. It takes as input a gray scale image, and produces as output an image showing the positions of tracked intensity discontinuities. The Canny operator works in a multi-stage process [40]. First of all the image is smoothed by Gaussian convolution. Then a simple 2-D first derivative operator is applied to the smoothed image to highlight regions of the image with high first order spatial derivatives. Edges give rise to ridges in the gradient magnitude image. The algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known as non-maximal suppression. This gives a thin output edge. Finally, hysteresis is used as a means of eliminating streaking.

Streaking is the breaking up of an edge contour in the output image, caused by the spatial derivative output fluctuating above and below the defined threshold. The tracking process exhibits hysteresis controlled by two thresholds: T_1 and T_2 with $T_1 > T_2$. Tracking can only begin at a point on a ridge higher than T_1. Tracking then continues in both directions out from that point until the height of the ridge falls below T_2. This ensures that edges are not broken up into multiple edge fragments.

We use the Canny edge detector to locate isopotential contours on the optically mapped image. This is described in further detain in Chapter 5.

2.5 Fingerprint Analysis

Fingerprint recognition is an important problem in computer vision and biometrics. Fingerprints have been widely used for personal identification [41]. The information carrying features in a fingerprint are the line structures called ridges and valleys. In Fig. 2.11 the black lines are the ridges while the white regions are the valleys.

As described in [42], fingerprint identification can be divided into three essential tasks, namely, fingerprint acquisition, fingerprint classification and fingerprint matching.

Fingerprints are acquired by the impression of fingers on different kind of sensors *e.g.* optical, thermal and capacitive sensors. The traditional technique of inked fingerprints is also in use. Fingerprint classification assigns a particular image into a certain category according to its global ridge and furrow configuration. This is essential for arranging fingerprint images in a database. Matching determines whether two fingerprint images are of the same finger.

A feature extraction process analyzes the fingerprint image to extract salient features for use in matching. Several fingerprint features are shown in Fig. 2.11. Local discontinuities in the ridge flow, called minutiae are typically used for this purpose. Ridge endings (points where a ridge ends abruptly) and ridge bifurca-

Figure 2.11: A typical fingerprint image.

tions (points where a ridge diverges into multiple branches) are examples of such points. There are distinct singularities in the flow too, like core and delta points [43]. These points are used for the classification operation. They have also been used as reference points for the alignment of different images of the same finger [44]. Feature extraction systems use image processing based enhancement methods to compensate for poor image acquisition [45].

Fingerprint matching techniques can be placed into two categories: minutiae-based and correlation based. Minutiae-based techniques first find minutiae points and then map their relative placement on the finger. However, there are some associated difficulties. It is difficult to extract the minutiae points accurately when the fingerprint is of low quality. Also, this method does not take into account the global pattern of ridges and furrows. The correlation-based method is able to overcome some of the difficulties of the minutiae-based approach. However, it has some shortcomings of its own. Correlation-based techniques are affected by image translation and rotation. Matching different sized (unregistered) minutiae patterns is another problem. These methods require the precise location of a reference point to register the images prior to matching.

2.5.1 Reference Point Detection

A fingerprint recognition system needs to determine the pose transformation between the input fingerprint and the template. This requires reference points for image registration (the alignment of different samples). The singular points of the ridge structure, such as core and delta points, are often used as reference points.

The Poincare index is widely used to detect the number of core and delta points in the fingerprint image [19], [46]. Poincare index at a pixel (i, j) is the summation of all differences in orientations of successive pixels along a square shaped curve centered around the pixel (i, j). The Poincare index at pixel (i, j) can be represented as:

$$Poincare(i,j) = \frac{1}{2\pi} \sum_{k=0}^{K=N} \triangle(k) \tag{2.35}$$

$$\text{where } \triangle(k) = \begin{cases} \delta(k) & \text{if } |\delta(k)| < \pi/2 \\ \pi + \delta(k) & \text{if } \delta(k) \leq -\pi/2 \\ \pi - \delta(k) & \text{otherwise} \end{cases} \tag{2.36}$$

$\delta(k)$ is the difference in the local orientation field between consecutive pixels along a closed digital curve with N pixels. The Poincare index is zero at non-singular points. The Poincare index is $1/2$ for a core shaped singular point while it is $-1/2$ for a delta shaped singular point.

However some fingerprint images might be just partial images, with the singular points left outside the imaged area. Another limitation is that these methods are very sensitive to the noise in the orientation field. A small perturbation in the orientation field may result in false singular point detection. Thus methods based on the Poincare index of the orientation field work well only for good quality fingerprint images. As described in [47], the point with the maximum curvature on a convex ridge is an apt choice for a reference point for all types of fingerprints.

A method for the maximum curvature reference point location is described by

Jain *et al.* in [20]. The orientation field is extracted and smoothed (low pass filtered) over a local neighbourhood. They used a 5×5 mean filter for the smoothing. An image containing just the sine component of the smoothed orientation field is considered (*i.e.* $\sin(\theta(i,j))$ where $\theta(i,j)$ is the angle of orientation field at pixel location (i,j)). This is followed by a multiresolution analysis of the difference of the sine image integration over two defined regions of the orientation field. The geometry of the regions for integration have been formulated empirically. The regions were designed to capture the maximum curvature in concave ridges and do not yield accurate results for arch type fingerprints. Rotation of fingerprint image also leads to an erroneous detection in this case.

Another algorithm for detecting of reference points has been proposed in [47]. First an orientation field is extracted and smoothed. This yields an orientation field which is continuous and smooth except for singular areas. The reference point detection is based on the hierarchical analysis of the orientation field coherence. The orientation field coherence indicates the consistency of the local orientation in a neighbourhood along the dominant orientation. Orientation coherence in high curvature areas is poorer than that in smooth areas. Thus, a point with local minimum coherence at both coarse and fine scales is identified as a reference point. This algorithm tends to be sensitive to the smoothing kernels used for smoothing the orientation field as well as on the size of the finest level block.

In this thesis, we have developed an algorithm based on the DHHD technique, which identifies the point with maximum curvature on the fingerprint image. Our method is robust to noise and to scaling and rotation of the image. The details of the proposed technique is presented in Chapter 4.

2.6 Summary

In this chapter we presented a comprehensive review of existing techniques for meteorological image analysis and fingerprint reference point detection. Techniques for locating critical points using motion fields were discussed. The deficiencies in these techniques were discussed which might be used to improve upon the current methods. We also provided a review of the techniques (level set method and DHHD) that are used in our algorithm. The work presented in this chapter has been accepted for publication [48].

Chapter 3

Review of Related Work: II

Ventricular fibrillation (VF) or cardiac arrest is a frenzied and irregular disturbance of the heart rhythm that quickly renders the heart incapable of sustaining life. The heart beat rate during VF can exceed 350 beats per minute (bpm), which is about 5 times the normal heart rate of 72 bpm. For a normal heart, its muscular contraction is smooth and coordinated due to a single wave of electrical excitation that signals the cells to contract. During fibrillation, higher frequency circulating activation waves are observed which cause small and out-of-phase localized contractions. Cardiac arrest causes nearly 250,000 deaths annually in the United States alone [21] and most of these sudden cardiac deaths are attributed to ventricular fibrillation.

Although a major problem in the modern world, the high frequency electrical activity occurring in the cardiac tissue during VF is still poorly understood. The only known therapy for VF at present is to apply a large electrical shock to the fibrillating heart and then wait for the heart to resume normal beating. Some consequences of the incomplete understanding of the phenomena, which caused failed antiarrhythmic drug trials are described in [21].

The mechanism of VF, and the characteristic electrical phenomena observed during VF are not fully understood and is an extensive area of research in the cardiac electrophysiology community. To gain a better understanding of the electrical wave propagation in the cardiac tissue, mapping techniques are the most common tool.

The concept of mapping rhythmic activation of the heart dates back to the beginning of last century. Initially mapping was performed using single probes to record activation in different regions of the heart. The 1960's and 70's saw the development of computerized mapping of the human heart. Most of the recent advances in cardiac mapping have focused on improving multisite recording techniques within the heart. These methods simultaneously record electrical activation from several hundreds of sites and have contributed significantly to the understanding of atrial and ventricular arrhythmia's. [3] provides a review of such techniques. However, the multisite contact mapping suffers from several limitations, such as the technical problems associated with amplification, gains, sampling rates and signal-to-noise ratio.

Motivated by the above-mentioned limitations of contact based mapping, the last few years have seen the development and use of voltage-sensitive dyes as a means to map cardiac activation. The technique is briefly described in Section 3.1. We describe the optical mapping apparatus and the role of cardiac optical mapping in understanding of various phenomena. Analysis of cardiac activation maps is discussed in Section 3.2 while Section 3.3 describes the application of models for the simulation of cardiac tissue. Finally we present an overview of the Canny edge detector in Section 2.4.

3.1 Cardiac Optical Mapping

An intact animal heart (such as frog, dog and pig) is extracted and pinned onto a dissection dish and stained with a voltage sensitive dye. Chemical motion inhibitors are used to reduce mechanical movement of the heart. The heart is paced using electrodes to induce different phenomena. The heart is illuminated using monochromatic light sources. The small changes in the fluorescent signal, due to excitation of the dye, is the signal of interest.

Voltage-sensitive dyes, when excited, provide an optical signal that mimics an action potential and thus allows the visualization of how electrical activation is propagating in any region under view. This allows for the precise visual evaluation of the propagation of a wave of excitation and measurement of its properties. Voltage-sensitive dyes bind to the cell membrane with high affinity and fluoresce light in direct proportion to transmembrane voltage. These dyes must be excited by light to induce fluorescence. The most common excitation light sources are tungsten-halogen lamps, mercury arc lamps and argon ion lasers. The precise shape of the emission spectra of the voltage sensitive dyes (and therefore the optical action potential) does not correspond to any absolute voltage, and only relative potential change is detected by this method. Optical mapping techniques use imaging devices, such as a photodiode array or a charge-coupled device video camera, to acquire the image sequences. A typical setup for optical mapping of cardiac electrical impulses is shown in Fig. 3.1.

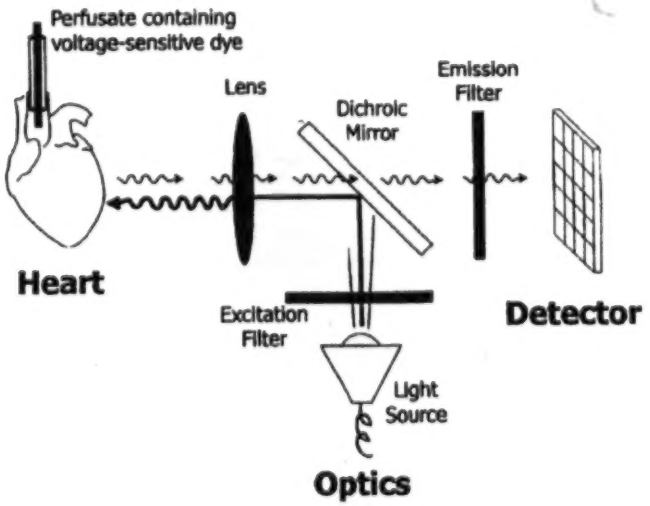


Figure 3.1: A cardiac optical mapping system (from [3])

Photodiode arrays and CCD cameras are the photodetectors used in these setups. The silicon photon diode detectors have a fast response time and signal acquisition

is relatively noise free because they have a low dark current (the amount of signal when no light is present). However, the number of photodiodes in an array is limited (about 50×50). To increase the sampling region, CCD cameras are used which provide higher spatial resolution of 128×128 pixels, as discussed in [49].

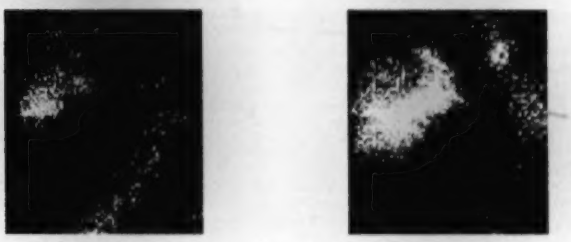
The emission filter and dichroic mirror serve as filters for monitoring the voltage sensitive dye used. The cutoff wavelength of the filter is dependent on the emission spectra of the dye.

Witkowski *et al.* [50] mention that commercial CCD cameras have a read noise level which is two orders of magnitude above that required to detect voltage sensitive dye intensity variations. Another significant issue is that the electrophysiological events in cardiac tissue occur in a time interval of the order of 1 ms. This calls for very high frame rates, if the optically mapped sequence of images is to capture the temporal detail of the event. Hence, direct use of commercially available products is not possible. An image acquisition system is described in [50], which achieves these specifications.

Fig. 3.2 shows video frames from optically mapped sequences. As can be observed, electrical impulses on a normal pig heart Fig. 3.2(a) tend to follow a regular pattern. During fibrillation, the electrical impulses are more haphazard as seen in Fig. 3.2(b). The ability to visualize the spread of excitation in intact hearts allows for comparisons between recorded dynamics and numerical simulations, as discussed by Christini and Glass [51].

Optical mapping was proposed with just surface recordings in mind. Newer techniques which can be used for visualizing the electrical activity from inside cardiac muscle via fluorescence measurements are described in [52]. Here, fluorescence measurements are performed in the transillumination mode, in which the light is placed behind the tissue and fluorescence emitted from the opposite side is collected. This is in contrast to the more common epi-illumination mode, in which

(a) Normal pig heart



(b) Pig heart undergoing fibrillation

Figure 3.2: Sample frames from the optically mapped video sequence.

the light source and photodetector are aimed at the same surface (as shown in Fig. 3.1).

3.1.1 Optical Mapping for Understanding Cardiac Arrhythmias

In optical mapping, the membrane potential is registered at every recording site over a sequence of time. Therefore, it is possible to relate complex propagation patterns to voltage changes occurring at the cellular and subcellular level. Studies performed at the level of the single cell have shed important insight into the nature of intracellular propagation within cells and as well as between cells. The role of cell-to-cell coupling, structural discontinuities, and tissue anisotropy in propagation of the electrical impulses were interpreted through cardiac mapping [53].

Application of optical mapping techniques to the study of cardiac fibrillation has given experimental proof of several theories of wave propagation in excitable media [3] such as the high-frequency reentrant sources that underlie fibrillation and generate spiraling waves that propagate throughout the ventricles in complex pat-

terns. Optical recording of the cardiac wavefront during reentry has demonstrated that conduction speed depends upon the curvature of the spiral wave, i.e., a greater degree of conduction slowing is noted in the presence of a more pronounced curvature.

Spiral waves were identified on the cardiac tissue, using optically mapped images of a fibrillating rabbit heart. Spiral waves occur in several systems, such as brain and retina. A particular perturbation of the excitation wave may result in spiral wave activity [23].

3.1.2 Application of Image Processing

In this section we review some of the image processing techniques that have been used in the context of cardiac optical mapping. Image processing techniques to enhance the quality of CCD images is a significant area of study. Common issues that can be tackled using image processing techniques are denoising of images, compensating for phase shifts and motion artifacts.

Phase shift: The CCD camera based optical imaging is more severely limited in temporal than spatial resolution. A phase correlation method is used in [54] to correct for the phase shift in time caused by propagation of the wave. This is followed by Gaussian filtering and median filtering for denoising. The phase correction before the spatial averaging leads to sharper features and better quality optical maps.

Denoising: A wavelet based image denoising technique was used by Witkowski *et al.* to improve the quality of the acquired optical map images.

Motion artifacts: Another application is for correcting of motion artifacts. One of the constraints of optical mapping is the deformation of the intensity and shape of optical potential due to cardiac muscle contraction (since the relative location of the tissue to the camera and light sources changes). Although chemical motion

inhibitors are used to prevent mechanical motion of the cardiac tissue, there is still some residual motion. Moreover, these chemicals are very expensive and hence alternative motion artifact cancellation techniques are an active topic of research. Motion artifacts due to cardiac motion are tackled in [55]. Image registration is used to align the sequence of digital images taken from the optical recorder. Optical alignment of a frame with the reference frame is obtained by maximizing a mutual information image similarity measure. This ensures that each location in the image represents an intensity measurement of the same tissue location throughout the recording procedure.

Locating singular points: Optical mapping videos have been analysed by Guo *et al.* [56]. Here, the optical flow field is extracted from the optically mapped video sequences. The DHHD algorithm is then applied to this extracted field to decompose the field into purely curl-free and purely divergence-free components as described in Section 2.2. These decomposed components are analysed for locating critical points (like sources, sinks etc.) in the motion fields. These critical points are essential for understanding the abnormal propagation of cardiac electrical signals. As an example, rotors (or spirals) in the cardiac tissue can be captured directly using this framework. The critical points detected by the algorithm were shown to match well with manual analysis of the video data.

The quantitative analysis of the cardiac videos have started receiving attention lately. Until now most of the existing techniques have been qualitative. Their analysis is dependent on the pattern recognition skills of cardiologist. We believe that the quantitative analysis of these videos will lead to significant improvement in understanding the phenomena of fibrillation and aim to analyze the videos using techniques from computer vision. We demonstrate the efficacy of the techniques proposed in [56] to the location of features of interest in the cardiac electrical patterns observed in the videos. This techniques are compared to the algorithm proposed by

[57] and are shown to perform better.

3.2 Analysis of Cardiac Activation Maps

One of the areas in the study of cardiac elctrophysiology is the visualization of cardiac arrhythmias. Isochronal cardiac activation maps are frequently used to study cardiac arrhythmias. They are constructed from local activation times associated with spatial locations. Each isochrone represents the leading edge of a cardiac activation wavefront at subsequent intervals of time. One significant disadvantage inherent in the isochronal maps is the excessive smoothing in regions of cardiac tissue that are sampled sparsely. To capture a particular feature of interest, the tissue has to be sampled at a finer resolution than the relevant feature.

Another parameter used to characterize wavefronts is conduction velocities, which contains information on the speed and direction of a propagating wavefront. These are measured using catheters (a tubular medical device for insertion into canals or vessels). An advantage of vector maps is that vectors measured using a catheter would only smooth data over a small area. At present, for clinical purposes cardiologists generally rely on isochronal maps to infer patterns in the cardiac activation sequence.

Fitzgerald *et al.* [58] performed a psychometric comparison of cardiac velocity vector mapping and isochronal mapping techniques. Comparative recognition of synthetic arrhythmia patterns presented as vector field and isochronal activation maps were investigated. The key goal was to study what types of maps provide the best quantity and quality of information with fewest number of measurements. It was shown that for simple arrhythmias, performance of cardiologists in selecting critical points was superior with vector maps to isochronal maps. For more complex arrhythmias, there was no significant difference in performance between vector and isochronal maps. However, arrhythmia features were clearer with vector

maps, proving that the velocity vector maps offer distinct insight into the clinical study of arrhythmias. These results show that a vector field representation can characterize propagation qualitatively and quantitatively, and is an effective alternative to isochronal mapping, especially in the case of complex rhythm patterns like those observed in fibrillation. (Conduction velocity vectors are not used in clinical practice at this time). In Section 5.2, we compare two velocity vector map analysis techniques.

3.2.1 Velocity Vector Maps

Velocity vectors, representing the velocity of potential on the cardiac surface are generally extracted using direct measurement on the cardiac surface with electrode arrays. The traditional method of calculating velocity is to first manually identify the direction of propagation and then measure location of an activation front and divide the distance traveled by the time interval. The inherent assumption here is that the wavefront travels in a direction perpendicular to the electrodes. However, this is not always the case, especially for complicated wavefronts, leading to spurious speed estimates. If meaningful insights are to be gained from velocity vector maps, accurate representation of the propagation of electrical waves is required. To address this issue, Bayly *et al.* [59] propose an alternative technique for the estimation of conduction velocity vectors. A polynomial surface $T(x, y)$ is fit to the space-time (x, y, t) coordinates of active points. (x, y) is the spatial location of an electrode site and t represents the time it was recorded as active. The fitted surface describes activation time as a function of position, *i.e.* each section is equivalent to an isochronal contour. The gradient is always normal to the isochrone and thus defines the direction of propagation. The speed and direction of propagation is now calculated using the gradient of the local polynomial surface. This polynomial surface method is more robust in dealing with missing or bad data from electrodes and

more robust to noise in the measurements. Morley and Vaidya [25] used the local velocity vectors calculated from the isochronal maps to illustrate the conduction properties of the mouse heart tissue during fibrillation.

Fitzgerald *et al.* [57] use velocity vector fields from cardiac mapping data for studying conduction features in cardiac tissue. These vectors are measured using intracardiac catheters. Mathematical operators such as the divergence and curl are used to analyze the vector fields and locate certain arrhythmia features without any need of human intervention (techniques based on isochronal maps rely on the pattern recognition skills of the cardiologist for locating arrhythmia features). Divergence of the vector field is used to locate extopic foci and wavefront collisions. The curl is used to identify central obstacles in reentrant circuits. The curl operator, however, fails to provide accurate localization for human arrhythmias. This study shows that velocity vector maps are a viable alternative (and possibly a better method) for studying cardiac activation.

3.3 Simulation of Cardiac Tissue

Another area of research is to understand the mechanism of electrical wave propagation in cardiac tissue [60] through simulation. Ionic models are used to simulate the cardiac tissue and study the propagation of electrical impulses there in. Distribution of membrane channels, the change in conductance of these channels *etc.* are some of the parameters included in the model. Such generic models of excitable media have been developed in 2- and 3-dimensions. Various electrical phenomena are simulated using these cardiac models to gain a better understanding of the electrical basis of cardiac phenomena like VF.

To give an idea about how the cardiac tissue is modelled, we present the model used in [24]. For an isotropic cardiac sheet, the transmembrane potential V_m is

given by:

$$\frac{\partial V_m}{\partial t} = D\nabla^2 V_m - \frac{I_{ion}}{C_m} \tag{3.1}$$

C_m is the membrane capacitance while D is the diffusion coefficient. I_{ion} is the total current flowing through the membrane, which is described using models for the sodium and potassium ion currents.

One such study demonstrated that the impulse propagation is dependent on tissue properties and the geometry of the excitation wavefront [4]. Wavefront curvature is an important property of spiral waves, which are known to cause some types of cardiac arrhythmias. The steady state velocity of a wave (V_0) is determined by the properties of the excitable tissue on which the wave is propagating. A first-order approximation of the relation between the curvature of a wavefront (K) and its propagation speed (V) has been derived for a continuous isotropic medium in [61] and is given by:

$$V(K) = V_0 - DK \tag{3.2}$$

Here D is the diffusion coefficient of the medium and is characteristic to a medium. When the excitation wavefront is curving outwards, the conduction velocity is lower than V_0 because the local excitatory current distributes over a larger membrane area. When the excitation front is curving inwards, the excitatory current converges in front of the propagating wave resulting in a velocity larger than V_0. Eqn. (3.2) also predicts that above a critical curvature $K_{cr} = V_0/D$, propagation will not occur. This is called conduction block. The basic mechanism relating wavefront curvature to velocity of propagation in an excitable medium is illustrated in Fig. 3.3.

The effect of curvature on wave velocity have been considered only rarely in heart tissue until recently. This is explained by the small radii at which wavefront curvature significantly affects conduction. Beaumont *et al.* [23] have proposed a technique to study wavefront morphology for arbitrarily shaped waves. -30 mV

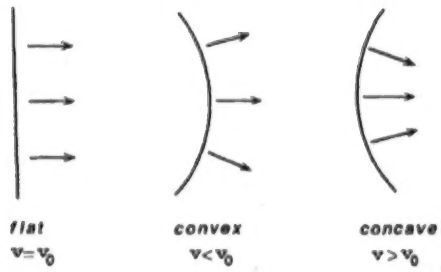


Figure 3.3: Effect of wavefront geometry on propagation velocity [4].

isopotentials of a spiral wave are parametrized by fitting a cubic smoothing spline to piecewise polynomial segments of the isopotential. This parametrized representation is used to calculate the curvature of the wavefront using standard differential geometric techniques. The point of zero curvature is used as a reference point to track the tip of the spiral as it meanders.

Kay and Gray [24] used voltage measurements to predict the curvature field in a wavefront. Their method is an extension of [23] for application to arbitrarily shaped wavefronts obtained from 2-D media. They locate isopotential points on the grid, fit these points to smooth splines to obtain a parameterized representation of the contour and then calculate the curvature using differential geometric properties of this contour. The use of splines is essential as the waves during arrhythmia are arbitrarily shaped. This framework provides a method to measure the curvature vector field of each individual wavefront within a dataset. These measurements are used for a quantitative analysis of the interaction between curvature and velocity.

Our aim is to combine the techniques used in studying electrical patterns in simulated models with image processing techniques in such a way that they can be useful to analyze cardiac optically mapped data. We present a technique to predict the curvature of an arbitrary shaped wave in a optically mapped image.

3.4 Summary

In this chapter, we presented a literature survey of the cardiac optical mapping method. We also discussed some image processing algorithms for processing optically mapped data. We then presented some techniques used to study properties of electrical waves using simulated cardiac tissue models and discussed how we intend to use techniques developed by such studies for the analysis of cardiac optically mapped data.

Chapter 4

Applications of the Discrete Hodge Helmholtz Decomposition to Image and Video Processing

In this chapter we explore two novel applications of the DHHD to image processing problems. Hurricane tracking is an important meteorological application. The eye of the hurricane represents a rotational center, which is shown to be robustly detected using DHHD. This is followed by an automatic segmentation and tracking of the hurricane eye using a level set based segmentation method, which does not require manual initializations. Results are presented using two actual hurricane videos.

We also consider the application of DHHD to fingerprint analysis. Identification of reference points, like whorls in the ridge structure, is an important problem in fingerprint matching. DHHD is used as a mathematically well founded and elegant method for this analysis. The new technique for reference point detection is relatively insensitive to noise in the orientation field. Our reference point detection method is not affected by the rotation and scaling of the fingerprint image. The DHHD based method is shown to detect reference points correctly for 96% of the images in the database used.

The subsequent discussion is organized as follows. A brief review of the literature on the application of image processing techniques to meteorology is given in

Section 4.1. Section 4.2 discusses motion estimation from a hurricane image sequence and the subsequent eye location and tracking results. Fingerprint analysis is described in Section 4.3. Details of the extraction of the orientation field and experimental results are also provided here.

4.1 Cloud Motion Tracking

Computer vision algorithms can be used to automatically interpret satellite images. This would lead to increasing the reliability of weather surveillance and could facilitate the assimilation of data from satellites for numerical weather prediction models. Image processing is widely used for the enhancement and analysis of remotely sensed data [62]. A review of some of the existing systems for image processing of meteorological images is presented here.

A major problem in meteorological data analysis is shape recognition and detection [63]. General techniques involve shape extraction followed by point correspondence [17], [64], [65]. Mukherjee and Acton [17] present a cloud tracking algorithm. Here the clouds are first segmented in the image using scale space classification methods. A set of feature points are then identified on the contour of the extracted cloud. Points of inflection on the contour are generally used as feature points. These inflection points are assumed to remain stable over a brief interval of observation and matching these corresponding inflection points between consecutive frames would yield the motion vectors. To establish a correspondence between feature points in consecutive frames, a cost function which enforces path and shape coherence is minimized. As mentioned in [66], feature matching is suitable for smaller sized images, but is not efficient for satellite images which span a large spatial area. For large scenes, several features have to be extracted within the image and matching them is computationally expensive. Also, identification of multiple feature points is a complex procedure on its own. Thus methods like [17] have a very

limited application and might not work for all kinds of remotely sensed images.

Systems for cloud tracking and characterization using satellite images have been developed. Rain clouds are tracked in [64] using algorithms based on the matching of morphological skeletons. Tracking and characterization of convective clouds is developed by Papin *et al.* in [67] and [68]. A level set segmentation method is used to extract the cloud shapes and to track them over a sequence of frames. Temperature measurements from infrared image sequences are available for the corresponding images and are used in the segmentation and characterization of clouds. Cloud characterization techniques are in general based on multi-spectral and textural information. A Bayesian estimation framework associated with Markov random field model is used to label the clouds.

Zhou *et al.* [30] proposed a scheme to perform structure and nonrigid motion analysis from 2D image sequences of hurricane images without correspondence. They use two image sequences of the same cloud structure (obtained from two different GOES satellites) to generate stereo information. Their technique is described in further detail in Section 2.1.4. These techniques have applications to meteorology in general. Weather prediction and cloud modelling are some of the areas where image processing techniques have been used.

In this thesis we present a scheme for tracking the eye of the hurricane over a sequence of frames. Only one set of 2D monocular images are used for the tracking. Our method does not require any other information about the event (*e.g.* temperature distribution) and just uses the visual channel of the satellite data for the tracking.

4.2 Proposed Scheme for Hurricane Tracking

Hurricanes are tracked by meteorologists by making use of various data like hurricane wind speed, the path the hurricane has followed over the past few days, satellite

tracking data and the pressure distribution in the region. Satellite data is particularly useful for the detection and monitoring of these storms over remote oceanic areas. The principal sources of data in addition to weather satellites are reconnaissance aircraft, coastal radars, and measurements from ships, buoys, and land stations. In this thesis, we present a method to track the hurricane using satellite images.

An overview of the proposed scheme is presented here. We extract the motion field in a hurricane sequence (from two consecutive frames) using the block matching algorithm. DHHD is applied to this motion field and is used to locate the rotational center, which is the eye of the hurricane. After eye detection, we track the motion of the eye over a sequence of frames. General segmentation techniques require manual initialization, after which the contour evolves to segment the desired shape. However, our method is automatic and does not require human intervention for initialization. This procedure for hurricane eye tracking does not require any form of feature extraction or matching, which is a significant advantage for satellite images that span a large spatial area.

The hurricane eye tracking between frames exploits the property of relative continuity in between frames. The location of the eye is assumed to be approximately constant between consecutive frames. The final segmented shape in a frame is used for providing the initial points for the level set segmentation in the subsequent frame. [30] developed a method to estimate accurately the nonrigid motion and the cloud structure in a hurricane. Our algorithm can locate the hurricane eye with only an approximate representation of the motion field, obtained using the simple Block Matching Algorithm (BMA). Hence, complex motion estimation for the nonrigid motion involved in a hurricane is not required.

4.2.1 Initial Processing

- *Smoothing:* Gaussian smoothing is performed on the images to remove un-

wanted noise.

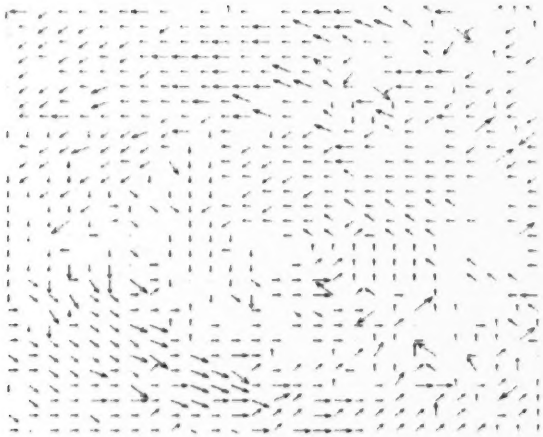


Figure 4.1: Motion field in a anticlockwise rotating hurricane sequence extracted using the BMA.

- *Motion Detection:* Two consecutive frames in a hurricane sequence are taken (I_n and I_{n+1}). Motion is estimated using the block matching algorithm (Section (2.1.1)). The image frame is divided into square blocks of a fixed size (8×8 in our case). The displacement of this block is detected by searching in a scan area (16×16) in the destination frame. The mean absolute difference is used as the searching criteria and is given by:

$$\delta = \min(\frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} |I_n(i,j) - I_{n+1}(i+u, j+v)|) \qquad (4.1)$$

In our implementation $m = n = 8$ and u, v range from -8 to 8. The block with the smallest difference δ is assumed to be the new position of the reference block. Motion estimation using the Horn-Schunck optical flow algorithm [5] and a affine motion model [69], [70] are also performed. However, block matching is the simplest of the aforementioned methods, and provides satisfactory results, so we just present the results obtained using this particular algorithm. Other algorithms have higher computational complexity [71]

and are generally not implementable in real time. The field extracted is shown in Fig. 4.1.

- *Motion Field Decomposition:* The DHHD is applied on the extracted field. The field is clearly not smooth and does not give an accurate prediction of the complex motion involved in the hurricane. However, our implementation of DHHD is quite robust and gives satisfactory results while locating the rotational center of this field, even with the rough estimate of the motion field. Since we are trying to detect a rotational center in this particular application, only the divergence-free component of the field is of importance to us. The divergence-free potential obtained is shown in Fig. 4.2. Since the video sequence shows the hurricane rotating in an anticlockwise direction, we get a distinct maxima. We need to locate the extremum points on this potential surface.

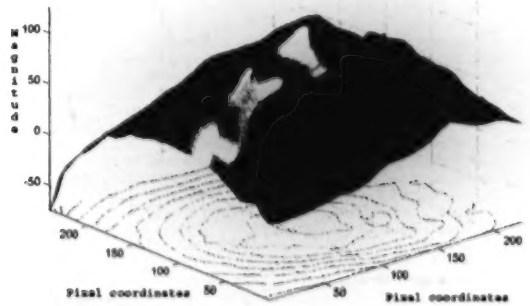


Figure 4.2: The divergence-free potential function with a distinct maximum and corresponding contours.

Locating the Extremum Point

The rotational center is located using the following property of the divergence-free potential function:

- A point is an anticlockwise rotational center if the potential surface has a maximum at that point;
- If the potential surface attains a minimum at a point, the point is a clockwise rotational center.

However, a simple location of minimum/maximum magnitude would yield erroneous results. This can be seen in Fig. 4.2, where the minimum magnitude does not correspond to a point of extremum. To locate the extremum point, the gradient of the divergence-free potential is calculated. For a continuous surface, we should have $\frac{\partial \vec{W}}{\partial x} + \frac{\partial \vec{W}}{\partial y} = 0$. Since we have a discrete grid, we might not get exactly zero at the extremum. Hence, we search for points which satisfy:

$$\left| \frac{\partial \vec{W}}{\partial x} + \frac{\partial \vec{W}}{\partial y} \right| < 4 \tag{4.2}$$

These points are the possible critical points. The value of 4 is chosen empirically. A very small value might cause some critical points to be left undetected whereas a very large value will cause far too many points to be considered as a possible critical point. More than one point satisfying Eqn. 4.2 is found. A point which has elements of opposite sign in all directions is chosen as the extremum point. An example of the gradient values at a point of extremum is shown in Table 4.1. This point represents a point within the eye of the hurricane.

-14.687	-6.5946	5.3749
-10.756	-2.2753	8.6554
-5.8657	2.0644	10.705

Table 4.1: Gradient values around an extrema point. Here -2.2753 is the extrema point.

To find out the direction of rotation (clockwise or anti-clockwise), the second derivative, $\frac{\partial^2 \vec{W}}{\partial x^2} + \frac{\partial^2 \vec{W}}{\partial^2 y^2}$ is considered. A positive value for the extrema point (minima) will indicate a clockwise rotation and a negative value (maxima) will indicate

anticlockwise rotation. Now, we describe the segmentation and tracking algorithms used.

4.2.2 Segmentation of the Eye

The level set algorithm has been discussed in Section 2.3. Here we use the level set algorithm to segment the eye of the hurricane, after the initial estimate to locate the eye has been obtained. The rotational center estimate from the DHHD is used as the seed point for the level set algorithm. The algorithm evolves from the initial seed point to extract the shape of the hurricane eye.

4.2.3 Tracking of the Eye

To reduce the computational time for our algorithm, we exploit the continuity of the image frames. Instead of performing DHHD to locate the initial seed point in every frame, we assume that the position of the hurricane eye would be relatively constant between consecutive frames. A point which has been detected to be within the eye in frame n is used as the initial seed point for the level set segmentation in frame $n + 1$. The precise details of the implementation used is given below.

4.2.4 Implementation and Experiments

Data Used

Image sequences of hurricane Luis (September 1995) and hurricane Isabel (September 2003), obtained from the NASA GOES (Geostationary Operational Environmental Satellite) satellites have been used for the experiments. Both sequences show the hurricane rotating in an anticlockwise direction. The Hurricane Luis sequence has a smaller eye compared to the Hurricane Isabel video sequence. The Luis sequence has 200 frames while the Isabel sequence has 83 frames. The frames are separated by about 12 minutes in time.

The images are first smoothed using a 4×4 Gaussian filter, and then the motion field is estimated using the block matching technique. This motion field is then fed into the DHHD subroutine and we extract the divergence-free field and the potential.

We assume the following convention for the signed distance function in our implementation of the level set method. Points inside the curve are assigned a negative distance while the outer points are positive. The initial estimate from DHHD (corresponding to the rotational center) turns out to be well within the eye of the hurricane. This extremum point indicated by the DHHD algorithm, and three other points in its neighborhood at a distance of 5 pixels each, are used as the initial points for the level set algorithm. As these points are inside the eye, we need to evolve the curve outwards. Hence, a fast marching method can be used (speed function is only in one direction). We initially use a narrow band fast marching technique for evolving the curve, which gives us a rough estimate of the boundary. The final step of the segmentation procedure is about 10 iterations of the original level set method, which gives us a smooth and accurate final curve. The results are shown in Fig. 4.3. N_{iter} iterations of the fast marching step give satisfactory segmentation of the eye. N_{iter} depends on the size of the eye. For the video sequence Isabel, which has a bigger eye, $N_{iter} = 550$ gives good results; whereas for the Hurricane Luis sequence, 300 iterations are sufficient.

The consecutive frames in the sequences do not have a significant burst of motion. Hence, the location of the eye can be assumed to be approximately the same in consecutive frames. The continuity is exploited in the initialization of points for the eye segmentation in the next frame. 3 points in frame N, which have a signed distance function less than ϕ_{max} are chosen as the initial points in frame $N + 1$. ϕ_{max} is an empirically chosen threshold which ensures that the initial points chosen are well within the boundary in frame N, and hence will be inside the eye in frame $N + 1$. Points which are in the eye of the hurricane but close to the boundary in

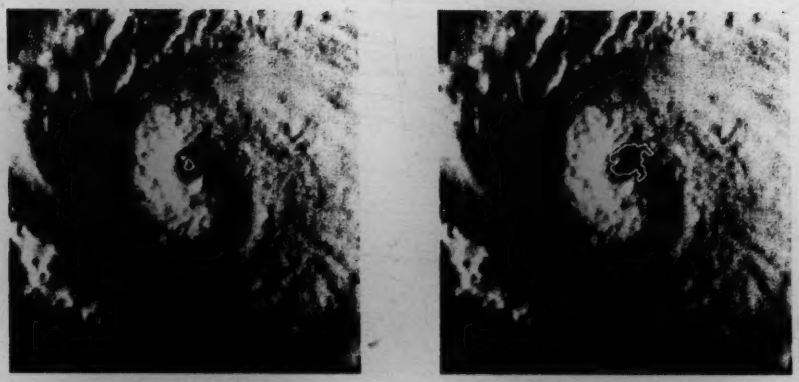


Figure 4.3: Level set extraction of the hurricane eye after 15 iterations (left) and after 300 iterations of the fast marching method(right).

frame N might not be inside the eye in frame $N+1$, thus leading to faulty initialization. The threshold ϕ_{max} makes sure that such points are not chosen as the initialization points.

ϕ_{max} depends on the size of the true boundary we are trying to segment. For the Isabel sequence $\phi_{max} = -7$, whereas for the Luis sequence (with a smaller eye), $\phi_{max} = -4.5$. Thus, we do not need to perform the DHHD step for every frame. Initialization points for the eye segmentation can be obtained using the continuity criteria, which increases the computational speed. However, this technique might fail if the discontinuity between frames is large. In the Isabel sequence, there is a large movement on every 9^{th} frame. So we perform DHHD to get new initial points once every 9 frames. The Luis sequence shows smooth transition between frames and DHHD is performed every 25 frames to obtain new initialization points.

The DHHD algorithm has been implemented in Matlab, while the level set tracking is a C program. For the 83 frames of the Isabel sequence, we call the BMA and DHHD 10 times and the level set program 83 times. The total time taken by BMA and DHHD combined is about 392 seconds while the level set program requires 649 seconds for its 83 calls. All these experiments have been performed on a Pentium 4 3.2 GHz machine with 1 GB of RAM.

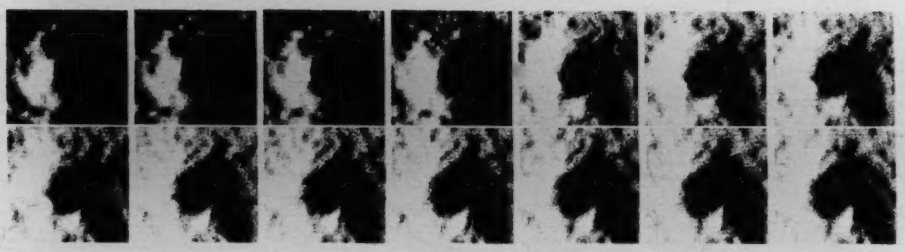


Figure 4.4: Hurricane eye tracking using level set segmentation: Sequence of images from the Hurricane Luis sequence, with the eye segmented.

A series of segmented frames are shown in Fig. 4.4. As can be observed from these sequences, the performance of the tracking algorithm is indeeed accurate. We have not found any work on hurricane tracking which is similar to this paper, and hence a comparative study has not been possible.

Every 20th frame of the segmented hurricane Isabel sequence is taken and averaged to generate Fig. 4.5. This shows the path the eye (and the hurricane in general) takes over a period of 17 hours.

Figure 4.5: Eye tracking over 100 frames

4.3 Algorithm for Fingerprint Reference Point Detection

DHHD is able to identify singular points in a motion field. Hence a point of maximum curvature can be detected in the orientation field of a fingerprint using our algorithm. We present a new algorithm for locating a reference point, which is defined as the point with maximum curvature in the ridge structure. Most of the methods based on the orientation fields are highly sensitive to the noise content in the field. We present a new method for reference point detection, which is robust to the noise in the orientation field. Moreover, our technique is not affected by the rotation or scaling of the fingerprint image.

Some preprocessing is initially done on the image followed by orientation field extraction. The reference point detection is accomplished by the application of DHHD on this orientation field. The steps involved are described in further detail below.

4.3.1 Preprocessing

The fingerprint image in consideration is preprocessed before analysis. An excellent source for fingerprint image enhancement is described in [45]. It discusses methods to improve the clarity of ridge structures of fingerprint images. The steps used in our implementation are:

- *Normalization:* An input fingerprint image is normalized so that it has a prespecified mean and variance. This process spreads the gray scale range of the given image over the entire spectrum of gray scales. Normalizing images makes it much easier to compare different images as all of them have the same range of gray scale. However it does not change the clarity of the ridge and valley structures. Normalization is a pixelwise operation and can

be described as:

$$I_{norm}(x,y) = \begin{cases} \mu_0 + \sqrt{\frac{\sigma_0^2(I_0(x,y)-\mu)^2}{\sigma^2}}, & if I(x,y) > M \\ \mu_0 - \sqrt{\frac{\sigma_0^2(I_0(x,y)-\mu)^2}{\sigma^2}}, & otherwise, \end{cases} \tag{4.3}$$

where $I_{norm}(x,y)$ is a pixel in the normalized image, $I_0(x,y)$ is a pixel in the original image, μ_0 and σ_0^2 are the desired mean and variance while μ and σ^2 are the mean and variance of the original image. We use $\mu_0 = 100$ and $\sigma_0^2 = 100$.

The effect of normalizing is show in Fig. 4.6.

(a) Original image (b) Normalized image

Figure 4.6: The result of normalization

- *Noise Removal:* Some images may be noisy. In the subsequent processing, we need to compute the intensity gradient of the image. This stage will be significantly affected by the amount of noise in the image. Hence, noise removal is a crucial step. A 4×4 median filter is used to filter the image. The orientation pattern estimated using a noisy image and its filtered version are shown in Fig. 4.7.

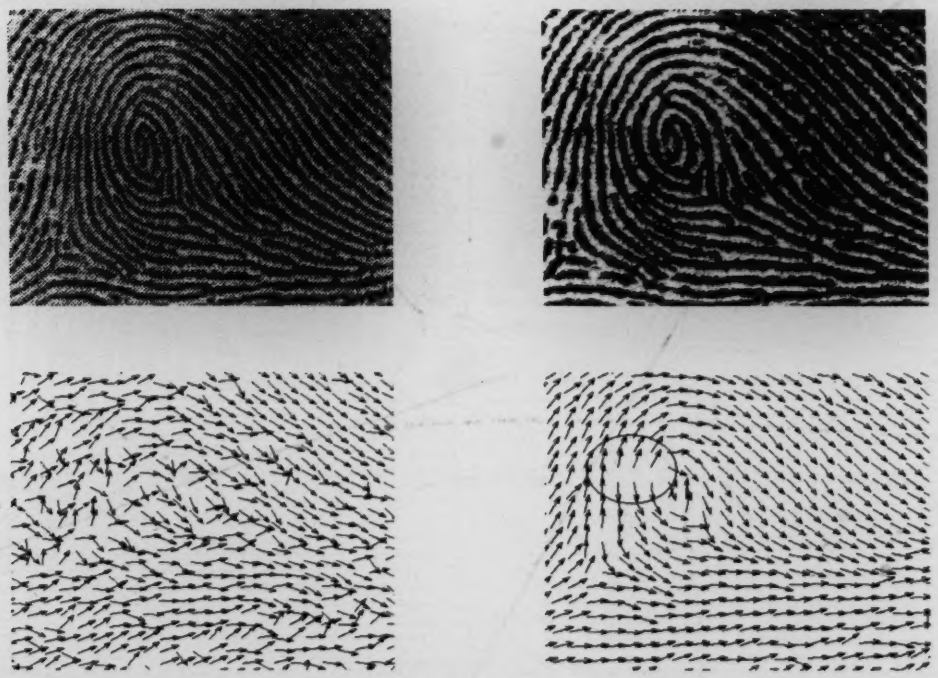


Figure 4.7: Orientation field extracted from a noisy fingerprint image (left) and from the same image after denoising (right).

4.3.2 Orientation Field Extraction

A directional field is used as a coarse representation of the fingerprint ridge structure and represents the local orientation of the ridge-valley structures. The direction field is in principle perpendicular to the gradients in a fingerprint image. They can be derived from the gradients by performing some kind of averaging operation on the gradients. The averaging is necessary because gradients are at a pixel level while direction field orientation is at a much coarser level. The averaging of gradients to obtain the directional field is discussed in [72].

First, the intensity gradient of the image is computed as:

$$\begin{bmatrix} G_x(x,y) \\ G_y(x,y) \end{bmatrix} = \text{sign}(G_x)\nabla I(x,y) = \text{sign}\left(\frac{\partial I(x,y)}{\partial x}\right) \begin{bmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial y} \end{bmatrix} \tag{4.4}$$

This definition of the gradient makes sure that the first element of the gradient vector is always positive. This convention is used to take care of the fact that in the

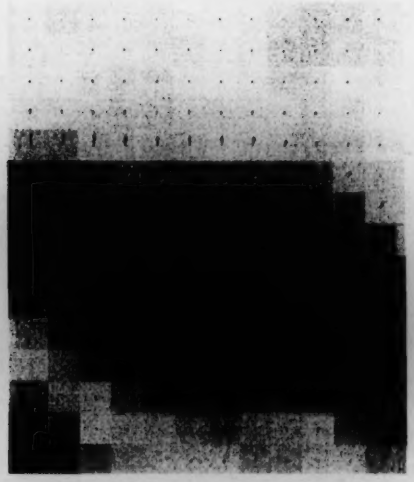


Figure 4.8: Gradient directions for a ridge

directional field, opposite directions indicate equivalent orientations.

The next step is to average the gradients over a block to obtain a single gradient direction for each block. However, a direct averaging of gradient vectors is not possible as gradients in opposite directions will cancel out upon summation. This is caused by the fact that local ridge valley structure remains the same if rotated by 180 degrees. The gradient orientation can be thought of being distributed in the cyclic space ranging from 0 to π. Hence the averaging is equivalent to calculating the π periodic cyclic mean.

A solution to this problem is to double the angle of the gradient vector before averaging. Since the gradient vectors that cancel out differ by π, doubling the angle ensures that all vectors are aligned in the same direction. Vectors which are perpendicular (angle between the vectors is $\pi/2$) are oriented in opposite directions upon doubling, and hence cancel out. The length of the vector is also squared before averaging. This doubling of angle and squaring of length can be represented mathematically as:

$$(G_x + j \cdot G_y)^2 = (G_x^2 - G_y^2) + j \cdot (2G_xG_y) \tag{4.5}$$

Eqn. (4.5) can be interpreted as if the gradient is a complex number and we are taking a square of a complex number. This is because doubling the angle and squaring the length of a vector is equivalent to squaring the complex number. The vectors obtained above are now averaged over a $W \times W$ block. We use $W = 16$ in this work. Hence, the average squared gradients $\overline{G_{sdx}}$ and $\overline{G_{sdy}}$ are given by:

$$\begin{bmatrix} \overline{G_{sdx}} \\ \overline{G_{sdy}} \end{bmatrix} = \begin{bmatrix} \sum_W (G_x^2 - G_y^2) \\ \sum_W (2G_x \cdot G_y) \end{bmatrix} \tag{4.6}$$

The average gradient direction is now given by:

$$\Phi = \frac{1}{2} \tan^{-1} \left(\frac{\sum_W (G_x^2 - G_y^2)}{\sum_W (2G_x \cdot G_y)} \right) \tag{4.7}$$

where $-\frac{\pi}{2} < \Phi < \frac{\pi}{2}$. The ridge valley field can be extracted from Φ as:

$$\theta = \begin{cases} \Phi + \frac{\pi}{2} & \Phi \leq 0 \\ \Phi - \frac{\pi}{2} & \Phi > 0 \end{cases} \tag{4.8}$$

Segmentation

A fingerprint image might contain regions with irrelevant information like dirt and smudges left on the acquisition device from previous acquisitions. The objective of segmentation is to locate the region of interest in the fingerprint image. The orientation field is randomly oriented in regions that do not contain the ridge structure and hence segmentation is important before further analysis. This can also be thought of as a *foreground/background* separation. We use a method of segmentation based on the concept of certainty level of orientation field estimation. This method was proposed by Jain *et al.* in [73].

After the orientation field is estimated, a certainty level of the orientation field at each pixel is calculated as follows:

$$CL = \frac{1}{W}\sqrt{\frac{\overline{G_{sdx}}^2 + \overline{G_{sdy}}^2}{\sum_W (G_x^2 + G_y^2)}} \tag{4.9}$$

Here we use the same notation as in Eqn. 4.6. The certainty level of the orientation field in a block W quantifies the extent to which the pixel gradient orientations agree with the block gradient orientation. For each pixel, if CL is below a certain threshold T_s, then the pixel is marked as a background pixel. A result of the segmentation method is shown in Fig. 4.9.

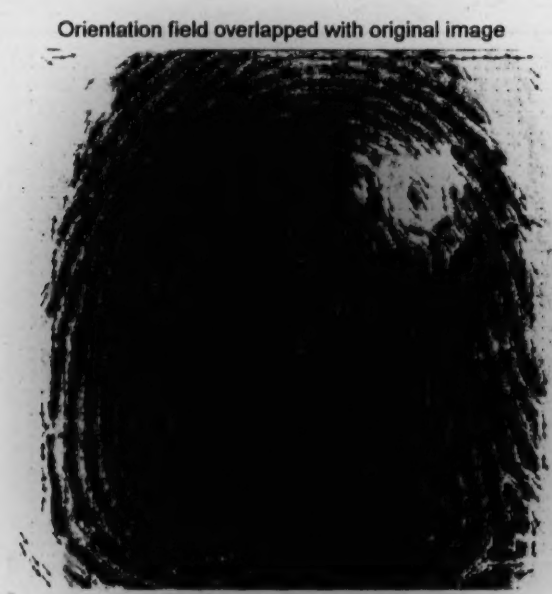


Figure 4.9: Orientation field superposed on the region of interest in a fingerprint image.

The range of θ, the angle representing the orientation field direction, is $\frac{-\pi}{2} \leq \theta \leq \frac{\pi}{2}$. Therefore, the extracted motion field has a discontinuity of π. This discontinuity in the orientation field can be observed in the circled part in Fig. 4.7 and in Fig. 4.10. This will cause DHHD to fail as the direction of flow is reversed abruptly.

To eliminate this discontinuity, the squared directional field is taken [74]. The

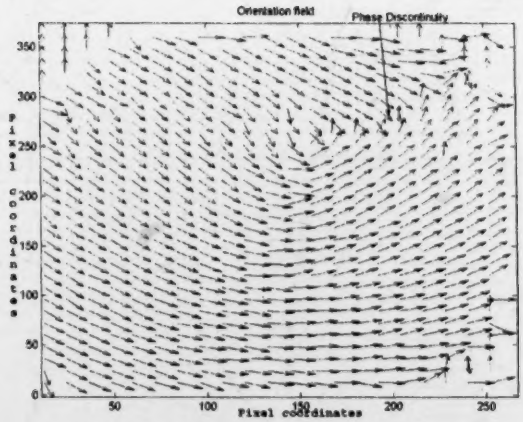


Figure 4.10: Discontinuity in orientation field for a fingerprint image.

motion vectors are now represented as:

$$\begin{aligned} \vec{\xi}_x &= Re\{(sin\theta + j \times cos\theta)^2\} = sin^2\theta - cos^2\theta = cos(2\theta) \\ \vec{\xi}_y &= Im\{(sin\theta + j \times cos(\theta))^2\} = 2 \times sin\theta \times cos\theta = sin(2\theta) \end{aligned} \tag{4.10}$$

which does not have the step of π.

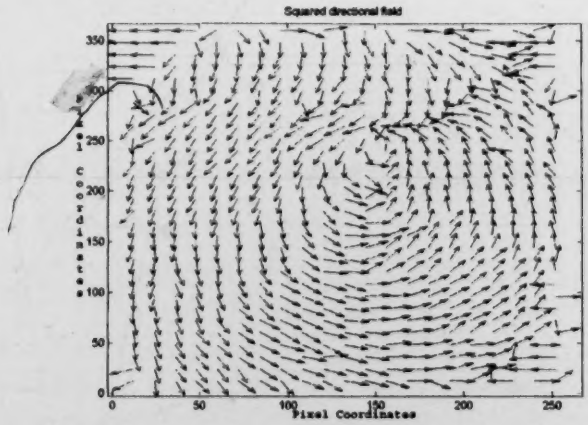


Figure 4.11: Squared orientation field, discontinuity is removed.

We are interested in the points with the maximum curvature in this orientation field. DHHD is now applied onto the squared directional field. We are trying to identify the point with the maximum curvature and hence only the divergence-free field and the potential are considered. The potential function should have an ex-

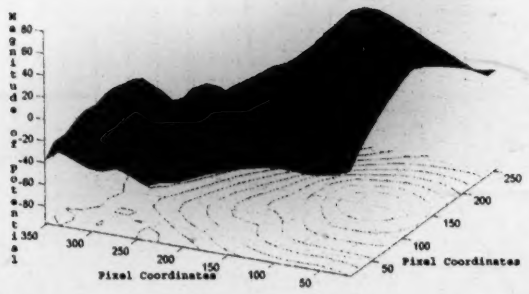


Figure 4.12: Divergence-free potential function and corresponding contours for a fingerprint image.

trema at the point of maximum curvature, hence identifying the point. This potential surface is shown in Fig. 4.12. To locate the extremum point, the gradient of the divergence-free potential is calculated as described in Section 4.2.1. An extracted reference point is shown in Fig. 4.14.

4.3.3 Experimental Results

Database used

The FVC2000 database is used to test our algorithm. It can be obtained in the DVD provided with [18]. The database has 4 different data sets of fingerprints, each collected using different sensor technologies. Details can be found in Table 4.2. Each data set has 8 different images of the same finger, and there are images of 110 fingers (880 fingerprints in all). We use the images of 50 fingers from each data set to test our methods. Further details about the database are described in Table 4.2.

	Sensor type	Image size	Resolution
DB1	Low-cost Optical Sensor	300 × 300	500 dpi
DB2	Low-cost Capacitive Sensor	256 × 364	500 dpi
DB3	Optical Sensor	448 × 478	500 dpi
DB4	Synthetic Generator	240 × 320	about 500 dpi

Table 4.2: Properties of the FVC2000 database

(a) DB1 (b) DB2

(c) DB3 (d) DB2

Figure 4.13: Sample fingerprints from the four databases.

Reference points, detected for different images of the same finger, are shown in Fig. 4.14. We use the same criteria as in [47] for performance measurement. The distance between the DHHD predicted reference point and the manually detected reference point is considered as the metric of the performance of our algorithm. The following criteria are used:

- If the detected point is within 10 pixels of the manually detected reference point, the localization is considered accurate.
- If the detected point is 20 pixels away, we assume detection with tolerable error.
- The detection is classified as an error if the distance between the detected and

manually identified points is greater than 20 pixels.

Our database consists of 500 dpi images. For a 500 dpi image, 20 pixels correspond to an 1 mm of physical distance.

Figure 4.14: Reference point identified in two different images of the same finger.

There are some image sets where the reference point is not included in the fingerprint. If the reference point is detected to be at the border, as shown in Fig. 4.16, it is considered to be a correct detection. If it is elsewhere, as in Fig. 4.15, a faulty detection is reported.

The results are shown in Table 4.3. We get correct detection in 96.25% of the cases, which is comparable to the 94.79% correct detection rate in [47]. There are 37 images where the reference point has not been imaged in the fingerprint.

Figure 4.15: Faulty detection of reference point, when actual reference point is not included in the fingerprint.

The algorithm has been implemented in Matlab. The entire algorithm (time includes preprocessing, field extraction and DHHD) takes 3.1 seconds on a 3.2 GHz Pentium 4 3.2 Ghz machine with 1GB of memory.

Distance (pixels)	Number of fingerprints	Percentage	Results from [47]
≤ 10	1355	84.69	81.07
> 10 and ≤ 20	185	11.56	13.72
> 20	60	3.75	5.21

Table 4.3: Distance between reference point predicted by DHHD and point predicted manually.

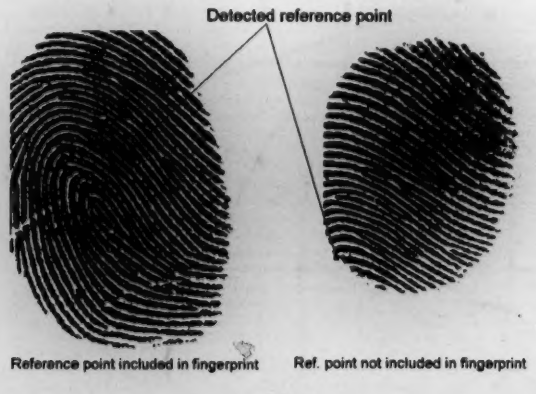


Figure 4.16: Images of the same finger with and without reference point, and the detected reference points.

4.4 Summary

In this chapter we presented an algorithm for the automatic tracking of the eye of a hurricane in a sequence of satellite images. The motion between two consecutive frames was estimated using a block matching algorithm and then DHHD was used to identify the center of rotation (the hurricane eye). The estimate of the eye was then used to initialize a level set algorithm which tracked the eye over consecutive video frames.

We also developed a technique for the robust detection of reference points in fingerprint images. DHHD is shown to be able to identify reference points in fingerprints. First the orientation field is extracted from the fingerprint image, after which DHHD is applied to the extracted field for critical point identification (the point with maximum curvature).

Chapter 5

Cardiac Video Analysis

In this chapter we present the overall framework used to analyze the cardiac videos. We start with a description of the cardiac optically mapped video sequence in Section 5.1. Velocity vector maps of electrical wave propagation on the cardiac tissue are gaining importance. We compare the method proposed in [56] to that discussed in [57] and point out cases in which the method developed in [56] is superior. This comparison is presented in Section 5.2. Techniques used to measure the curvature of a wavefront in the cardiac video are described in Section 5.3. An approximation to ECG signals can be extracted from the optically mapped video. This is discussed in Section 5.4. Finally a summary of the chapter is presented in Section 5.5.

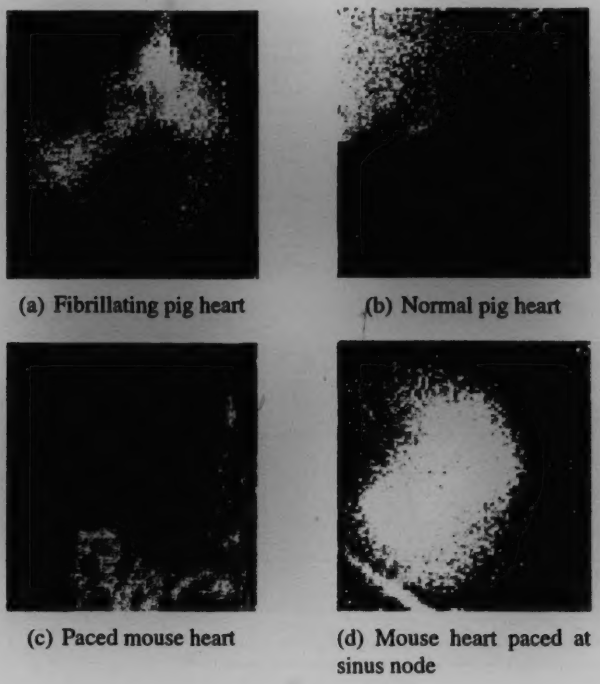
5.1 Cardiac Video Description

Details of the system used to obtain the videos which we are using can be found in [50] and have been described in Section (3.1). The characteristics of the four videos used for analysis have been briefly described in Table (5.1). To give an idea about the area of cardiac tissue imaged, $25cm^2$ corresponds to approximately 25% of the epicardial surface of a pig heart. Sample frames from each video have been shown in Fig. 5.1. The setup used for obtaining such videos is shown in Fig. 3.1. The videos are obtained using di-4-ANEPPS as the voltage sensitive dye and light sources of 500 ± 40 nm while a filter passing wavelengths above 590 nm is

used. The voltage sensitive dye is injected into the heart to observe the propagation of the electric waves. Voltage sensitive dyes are good probes for the detection of millisecond membrane potential changes and hence provide an excellent method to observe cardiac electrical activity with good temporal and spatial resolution.

	Video 1	**Video 2**	**Video 3**	**Video 4**
Subjects	Left ventricle of an isolated mouse heart	Left ventricle of an isolated rat heart	Left ventricle of an isolated pig heart	Left ventricle of an isolated pig heart
Phenomena observed	Pacing	Pacing	Normal beating	VF
Video length (frames)	999	2733	999	2303
Frame rate (fps)	Unknown	838	838	838
Image Size (pixels)	157 × 146	157 × 146	150 × 132	157 × 159
Target area (cm^2)	Unknown	1 × 1	5 × 5	5 × 5

Table 5.1: Description of videos



(a) Fibrillating pig heart

(b) Normal pig heart

(c) Paced mouse heart

(d) Mouse heart paced at sinus node

Figure 5.1: Sample frames from different videos

One essential point regarding the video is its frame rate. The video was captured using a CCD camera operating at 838 frames per second. Such a high frame

rate is essential to capture the temporal details of the electrical activity. For the motion estimation from the video sequence, we use the optic flow algorithm proposed in [5] and discussed in Section (2.1). The standard mathematical model used to find the optic flow is based on the brightness constancy assumption (Eqn. 2.3). The high frame-rate video sequence, which we are using, does actually satisfy this assumption. In other video applications, the violation of the brightness constancy assumption is common and is generally handled using robust statistics [8]. However, a simple optical flow implementation serves our purpose.

5.1.1 Preprocessing

As can be seen from the sample frames in Fig. 5.1, the images are really noisy. Some standard image processing techniques are used to improve the image quality prior to further processing.

Denoising comprises of smoothing the images to remove the noise. However it is important to avoid distorting the boundary of the waves as we will perform a curvature analysis (curvature of a wavefront depends on the shape of the wave contour). Simple Gaussian filtering tends to blur the edges in an image. To prevent this, we use a Wiener filter which performs linear adaptive smoothing based on the local image variance (the regions with a larger variance are smoothed to a lesser extent). This helps in preserving edges and other high frequency features in the image. A brief description of the operation of a Wiener filter follows.

The filter estimates the local mean (μ) and variance (σ^2) around each pixel in an $N \times M$ neighbourhood η [75]:

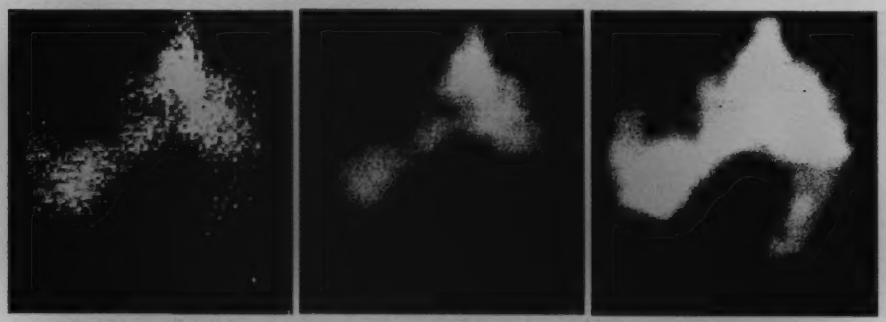
$$\mu = \frac{1}{MN} \sum_{n_1,n_2 \in \eta} I(n_1, n_2) \quad ; \quad \sigma^2 = \frac{1}{MN} \sum_{n_1,n_2 \in \eta} I^2(n_1, n_2) - \mu^2 \qquad (5.1)$$

The Wiener filtered image is given by:

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - \nu^2}{\sigma^2}(I(n_1, n_2) - \mu) \qquad (5.2)$$

Here ν^2 is the noise variance which is estimated as the average of all the local variances.

Temporal averaging is not used here as it was observed that this operation blurred the waveform shape. After smoothing using a Wiener filter based on the statistics estimated from a 5×5 neighborhood, the contrast in the image is enhanced using histogram equalization. The histogram equalization operation transforms the input image such that the histogram of the gray level values of the output image matches a desired histogram. Further details can be found in [76]. Results of preprocessing are shown in Fig. 5.2.



(a) Original image (b) After image smoothing (c) After image smoothing and histogram equalization

Figure 5.2: Preprocessing results.

5.2 Comparison of Techniques for Cardiac Velocity Vector Analysis

Cardiac activation maps were discussed in Section 3.2. Here we focus on techniques for the analysis of velocity vector maps. Guo *et al.* [56] estimated a motion field from a video sequence of an optically mapped heart. The vector field obtained corresponds to the cardiac action potential velocity map. Previous studies of cardiac vector maps used data obtained from direct measurement on the cardiac tissue ([59]) or derived the vector maps from the isochronal maps ([25]). The analysis of a

cardiac optical map to extract the velocity vector field is a novel approach.

A sample motion field, extracted using the optic flow motion estimation algorithm, is shown in Fig. 5.3. This represents a vector map of cardiac electrical activity, and can be used to analyze various features of conduction.

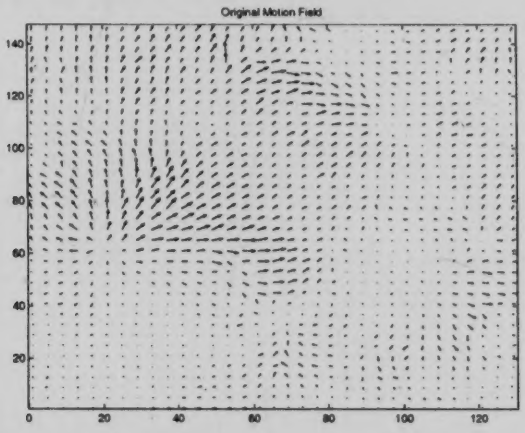
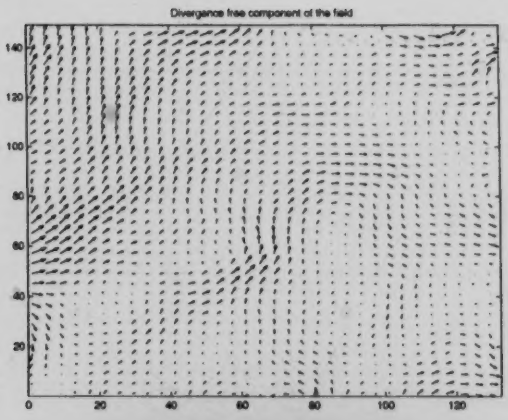


Figure 5.3: Original motion field extracted from video sequence.

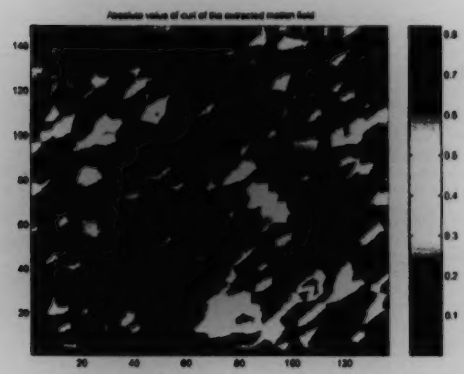
In [56], the estimated motion field is decomposed into its curl-free and divergence-free components using DHHD. The divergence-free component of the field in Fig. 5.3 is shown in Fig. 5.4(a). The analysis of the corresponding potential functions (described in Section 2.2) is then used to predict the critical points in the motion field. This technique has been used to identify singularities in cardiac videos in [56]. The critical points detected using the DHHD based algorithm match well with manual analysis of the video data.

Although it is difficult to visually distinguish the purely rotational component (i.e., the divergence-free component) from the input motion field (Fig 5.3), the decomposed divergence-free component (Fig. 5.4(a)) clearly shows that the pure rotational component in the pacing signals cannot be ignored.

We applied the algorithms proposed by Fitzgerald *et al.* [57] to the velocity vector field extracted from the cardiac videos to perform a comparitive analysis of



(a)



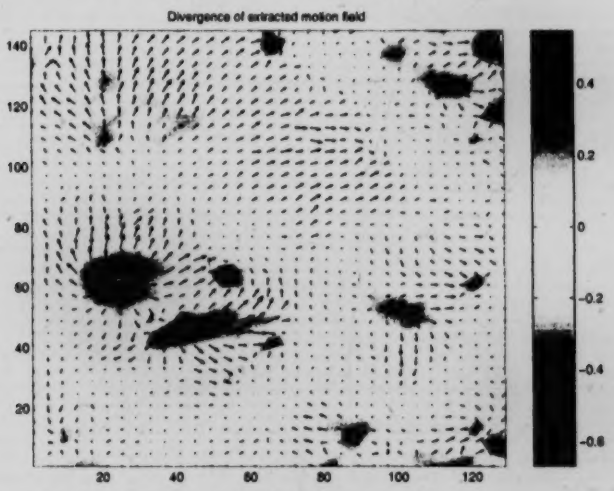
(b)

Figure 5.4: (a) Divergence-free component of original motion field. (b) Magnitude of the curl of the original field, shown as a color map.

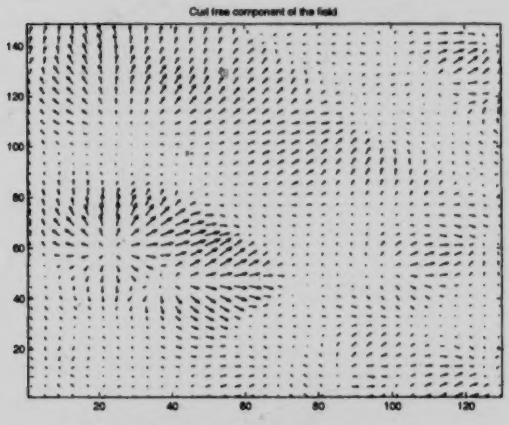
the two vector map analysis techniques. The divergence operator is applied to the extracted motion field (Fig. 5.3). The divergence is shown as a color map in Fig. 5.5. The red areas corresponds to areas of maximum divergence. These are points on the cardiac tissue from which electrical waves are originating. Such points are called ectopic foci. The color of the region gives a distinct picture of the size of the area in question. As can be seen in Fig. 5.5, both the curl-free component of the original field and the divergence of the original field are able to locate sources and sinks in the motion field.

We see that the critical points corresponding to the curl-free component are

located by both the techniques. However in this case the technique proposed by Fitzgerald *et al.* [57] has a slight advantage. It uses a simple divergence operator on the velocity field to predict the critical points. The divergence operator is computationally much cheaper than performing the DHHD decomposition and then analyzing the resulting potential function to locate the critical points.



(a) Divergence shown as color map



(b) Curl-free component

Figure 5.5: (a) The divergence of motion field shown as a color map with the curl-free field obtained using DHHD is superimposed on the color map. (b) The curl-free component of the field.

It is stated in [57] that the curl operator applied to the vector field is relatively

ineffective. The curl of the motion field is shown in Fig. 5.4(b), while the corresponding divergence free field obtained using DHHD is shown in Fig. 5.4(a). Simple visual analysis reveals rotational centers for the divergence free field, while the rotational features cannot be observed in the curl image. Taking this into consideration, the DHHD based method has a distinct advantage over the simple divergence/curl based method. Although computationally more expensive, it can locate critical points in both the rotational and the irrotational parts of the field. The rotational critical points provide important clues for describing and understanding the abnormal propagation of the cardiac electrical signals and hence are very important for the understanding of cardiac dynamics.

5.3 Wavefront Curvature Measurement

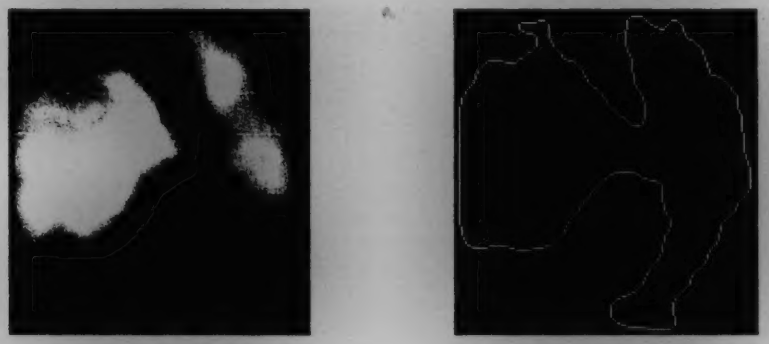
The study of the mechanisms of initiation and dynamics of spiral waves in myocardial tissue are an important research area. Such studies provide a better understanding of the origin and development of the most dangerous cardiac arrhythmias and eventually lead to therapeutic advances. Wavefront curvature is an important parameter in cardiac arrhythmias. The wavefront curvature determines the velocity of propagation of the wave. This phenomenon was discussed in Section 3.3. Convex waves are known to propagate slower than concave waves as was shown in Fig. 3.3.

5.3.1 Extracting Isopotential Contours

Isochronal maps are contour maps in which each contour gives the location of the wave front at constant time intervals. Isopotential contours, on the other hand, join all points on the cardiac tissue which are at the same electrical potential at a particular instant of time. We need to estimate the isopotential at each frame of a cardiac optically mapped video. The extraction procedure is as follows. Given the cardiac videos, we perform the preprocessing as described in Section 5.1.1.

Figure 5.6: Preprocessed frames.

A global image threshold is calculated using Otsu's [77] method. The threshold divides the image histogram into two regions such that the inter region variance is maximized. This threshold is then used to convert the graylevel image into a binary image. Now the Canny edge detector (Section 2.4) is used to extract thin edges from the image. These edges are an approximate representation of an isopotential contour on the cardiac surface. A frame from the optically mapped video of a fibrillating pig heart and its corresponding isopotential is shown in Fig. 5.7.



(a) Processed image (b) Extracted isopotential

Figure 5.7: A sample frame from optically mapped video and its extracted isopotential.

We need to estimate the curvature of these isopotential contours. The extracted

isopotential contour is comprised of discrete pixels and a direct estimation of curvature from the pixel values itself will lead to an inaccurate estimation of the curvature. Parameterizing the contour prior to using differential geometric techniques to calculate the curvature yields a much better estimate of the curvature. Once the discrete pixel coordinates, representing the contour nodes have been identified by the Canny edge detection method, we fit cubic smoothing splines to the node points to obtain a parametric representation of the contour.

5.3.2 Parameterizing the Isopotential Segments

The Canny edge detection algorithm returns a list of the coordinates of the nodes which belong to the isopotential contour. However the list is in raster format (*i.e.* each point in the edge is defined by its pixel coordinates) and needs to be rearranged in a sequential order of spatial contiguity before we can fit splines. Another goal is to assign a distance function to each node, indicating its relative position in the segment. We start with the first point, say p_0 (the point marked "initial point" in Fig: 5.8), in the list returned by the Canny detector and assign a distance of 0 to this initial point. A search for eight connected neighbours is performed among the points belonging to the isopotential edge, starting from the point to the immediate right of p_0. The search for neighbours is performed in the clockwise direction. The first detected neighbour is added to a new list, called *ArrangedList*, and the distance is incremented by one if it is 4-connected; otherwise, the distance is incremented by 2. Thus *ArrangedList* grows in "direction 1" in Fig: 5.8. The point is then removed from the list of isopotential node points. This process is continued until we cannot find a neighbouring point (*i.e.* when we reach "Point 2" in Fig: 5.8).

Once this happens, the search returns to the initial point p_0 and searches for neighbours again. This is to take into account contours that might be broken at one end (the nodes lying along "direction 2" in Fig: 5.8). For each node which is

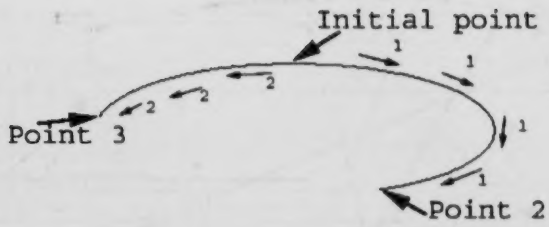


Figure 5.8: Rearrangement of contour points.

identified in this loop, it is assigned a negative arc length (progressively decreasing as we move away from p_0). This search for neighbours is continued until we cannot find any new neighbouring points (*i.e.* when we reach "Point 3" in Fig: 5.8).

A new segment is started after this and we repeat the above process. This is continued till all points in the original list have been considered. Segments with too few nodes are removed from *ArrangedList*. In our implementation, all segments with fewer than 25 nodes are eliminated.

The x and y coordinates of the isopotential nodes identified by the procedure described above are stored in a list $[x_i, y_i, u_i], i = 1 \ldots N$. Here u_i is the cumulative arc length.

Isopotentials are arbitrarily curved, especially during complex rhythms, where they may have long, straight sections as well as highly curved sections. We would like to emphasize here that while performing the spline fitting, sections with higher curvature require a greater tolerance than for sections which are relatively straight. A region of higher curvature (higher curve complexity) is shown in boxed region of Fig. 5.9. We used the technique described in [24] for estimating the local complexity of the curve. For each node, a square window is defined around the node. The number of isopotential nodes within this square K_n is used as an estimate for the complexity of the local shape. The tolerance of the spline fits is set to $\sqrt{0.05K_n}$.

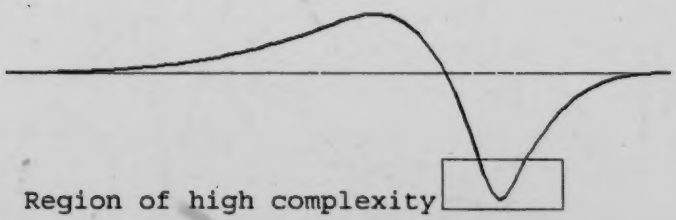


Figure 5.9: Demonstration of regions of curve with higher complexity.

To further minimize the error of fit for a segment centered at a node p, a weight is assigned to each segment node as described in [24]:

$$w(i) = \frac{\theta - (u_i - u_p)^2 + 1}{max[\theta - (u_i - u_p)^2 + 1]}; i = 1, \ldots, T_n \tag{5.3}$$

Here T_n is the number of consecutive node points which are used to obtain the spline representation *i.e.* solve Eqn. 5.5. θ is defined as $\theta = max[(u_i - u_p)^2]$. The parametric representation $\mathbf{C}$ of a segment of nodes (from node $i = 1$ to $i = N$) is given by the vector:

$$\mathbf{C} = \vec{\mathbf{X}}(u) : u_1 \leq u \leq u_N \tag{5.4}$$

Here $\vec{\mathbf{X}}(u) = [x(u), y(u)]$ are the continuous functions that give the indices at any location on the segment. Cubic smoothing splines [78] are fitted to the list of ordered node segments (x_i, y_i, u_i) to generate $x(u)$ and $y(u)$. A spline is fit to the points u_i, x_i to generate $x(u)$. Given the data points (u_i, x_i) with $u_i = [u_1, \ldots, u_N]$ a smoothing spline minimizes:

$$\rho \sum_{i}^{T_n} w_i |x_i - x(u_i)|^2 + \lambda \int_{u_1}^{u_N} |D^m x(u)|^2 du \tag{5.5}$$

Here ρ is the smoothing parameter while w_i are the positive weights and D^m represents the m^{th} derivative. $x(u)$ are the node points on the best fit spline. In Eqn. 5.5, the left term represents an error measure between the data points and the predicted points while the right term is a roughness measure. The minimization approach used in [79] is to make the roughness measure as small as possible subject

to the condition that the error measure is no bigger than the prescribed tolerance. The $y(u)$ coefficients are formulated similarly.

A sequence of cardiac optically mapped frames are shown in Fig. 5.10. In each frame, the parametrized spline representation of the isopotential is superposed on the image. The parametrized representation is smooth (unlike the contour obtained from the Canny operator) and hence compatible with applying differential geometric operators to calculate the curvature.

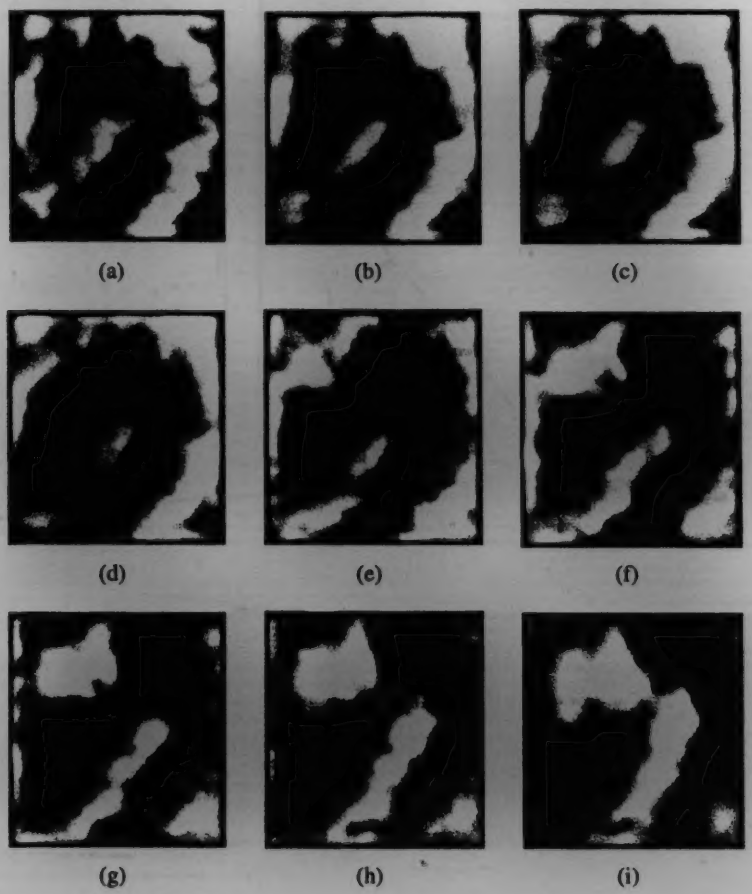


Figure 5.10: Detected isochronal segments in a sequence of frames. Spline approximation of the segmented contour are superposed on the corresponding frame.

5.3.3 Curvature Calculation

Once a parametrized representation of the isopotential segment has been obtained, the curvature can be calculated using standard differential geometric techniques [80]. The tangent of a parametrized segment is given by calculating its normalized first derivative with respect to its arclength:

$$\vec{T}(u) = \frac{d\vec{X}(u)/du}{|d\vec{X}(u)/du|} = T_x(u)\hat{i} + T_y(u)\hat{j} \tag{5.6}$$

The curvature vector is the derivative of the tangent $\vec{T}(u)$ with respect to the arclength:

$$\vec{K}(u_p) = b\left(\frac{d\vec{T}(u_p)}{du}\right) = b(K_x(u_p)\hat{i} + K_y(u_p)\hat{j}) \tag{5.7}$$

where u_p is the arclength at node p and b is the scale factor for normalization in the central difference ($b = 1/\triangle x$, and $\triangle x$ is the physical distance between two pixels). The curvature vector points to the center of curvature. Curvature is calculated in this way for each node along the isopotential. A frame with its parametrized isopotential and curvature vectors marked on the isopotential segment is shown in Fig. 5.11.

Figure 5.11: Curvature vectors calculated from parametrized isopotential.

5.3.4 Calculation of Velocity

We use the methods described by Kay and Gray [24] to obtain a measurement of the wavefront velocity. The first step for this is to identify which nodes of the isopotential belong to the wavefront. In terms of the node voltage V_m, nodes with a positive dV_m/dt belong to a wavefront. Carrying over this analogy to the intensity of a pixel (the intensity of a pixel is proportional to the voltage at that pixel), we use the following criteria to classify a pixel on the isopotential as a wavefront node: $I_t(m,n) - I_{t-1}(m,n) > 0$ and $I_{t+1}(m,n) - I_t(m,n) > 0$. Since spatial smoothing has been performed in the preprocessing step, we do not need to consider smoothing before calculating the difference. However, we use the two difference measures to increase the robustness of wavefront node detection. Unless both terms are positive, a node is not classified as belonging to the wavefront.

To detect the velocity, we use the conventional assumption that wavefront velocity is normal to the wavefront. The normal to an isopotential at node p is given in terms of the tangent unit vector of the parametrized segment at node p:

$$M = \frac{-T_x(u_p)}{T_y(u_p)} \qquad B = -Mx(u_p) + y(u_p) \tag{5.8}$$

where M is the slope and B is the intercept of the normal. $x(u_p)$ and $y(u_p)$ are obtained by evaluating Eqn. 5.4 at u_p.

To calculate the velocity, we consider the isopotentials of two frames n and $n+1$. The isopotential in frame n is parametrized using cubic splines and its normal is calculated. Nodes belonging to the wavefront are identified in the isopotential segments of both frames. For each wavefront node in frame n we search along the direction of the normal for corresponding nodes belonging to the wavefront of frame $n+1$. If such a point is found, the velocity is calculated as:

$$\vec{V} = \frac{(x_{n+1} - x_n)\Delta x}{\Delta t}\hat{i} + \frac{(y_{n+1} - y_n)\Delta y}{\Delta t}\hat{j} \tag{5.9}$$

(x_{n+1}, y_{n+1}) and (x_n, y_n) are the corresponding wavefront nodes in frame $n+1$ and frame n. $\triangle x$ and $\triangle y$ are the x and y physical displacement between neighbouring pixels and $\triangle t$ is the time between two consecutive frames. However in certain cases, there might not be a one to one matching between points, *i.e.* due to deformation of the wave the movement along the perpendicular assumption might be violated.

The curvature of such a point is calculated using Eqn. 5.7. Once all nodes on the wavefront have been processed, we plot the velocity versus curvature values. Fig. 5.12 is an example of such a plot.

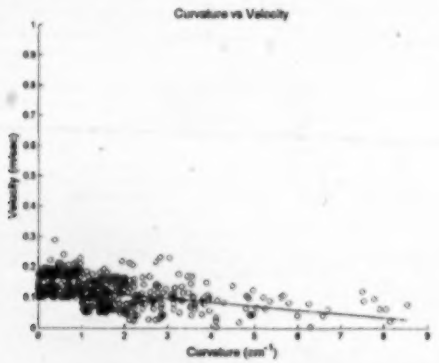
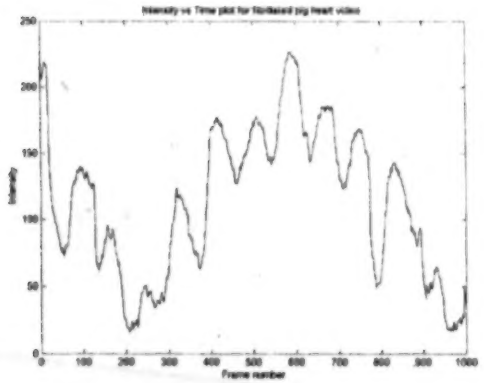


Figure 5.12: Wavefront velocity vs. wavefront curvature.

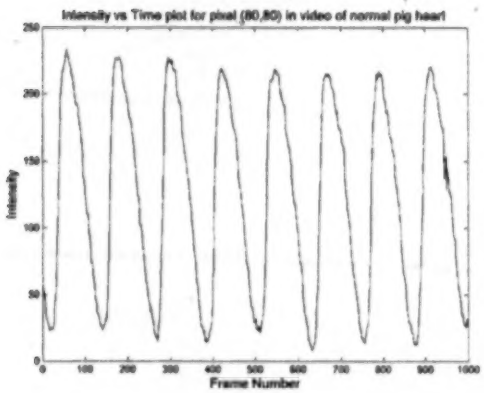
A straight line is fit to the data in a least squares sense as shown in Fig. 5.12. We get the following equation for the line:

$$V(K) = 0.13 - .0077K \tag{5.10}$$

These detected parameters are quite close to the actual parameter values for cardiac tissue. In [24], a typical value of V_0 is mentioned to be $0.39m/s$ and a diffusion coefficient $D = 7.6 \times 10^{-}3cm^2/ms$. The close match in detected parameters proves the efficacy of our technique for the analysis of wavefront morphology from cardiac optical maps.



(a) Fibrillating heart



(b) Normal heart

Figure 5.13: Intensity vs. Time plot at a single pixel location for a pig heart.

5.4 Information Extraction from Optical Maps

We developed some other algorithms which might help in the interpretation of optical maps. One of them is to estimate the action potential variation with time at a particular location on the cardiac tissue. Since the pixel intensity is related to the action potential at that location, we plot the pixel intensity over time to estimate the action potential variation. Since the images obtained are very noisy, this plotting of intensity values should be done after the images have been smoothed. Figs. 5.13(a) and 5.13(b) show the pixel intensity vs. time plot for a fibrillating and a normal

heart, respectively. It is clear that the optically mapped data allows clear estimation of the main action potential characteristics. The heart rate for a normal heart can be obtained by analyzing Fig. 5.13(b). Fig. 5.13(a) shows an irregular pattern, corresponding to the irregular heart beats when a heart is undergoing fibrillation.

5.5 Summary

In this chapter we presented a framework for analyzing cardiac videos. Initially an overview of the image enhancement operations for the optical maps was described. Kernels which retain the edges of images while performing the necessary smoothing are preferred. Two different methods for analyzing cardiac velocity vectors are compared and their pros and cons are described. We compared the performance of the two techniques on velocity vectors extracted from cardiac video frames and concluded that the technique proposed in [56] is more effective than the methods given in [57]. Finally a scheme to perform wavefront morphological studies on cardiac wavefronts is developed. A method to accurately estimate the curvature of an isopotential on the cardiac video image is developed. This is followed by calculation of the velocity of propagation of the wave on the cardiac video. The results from the curvature-velocity analysis are quite accurate in predicting the physical parameters of the cardiac tissue.

Chapter 6

Conclusions and Future Work

In this thesis, we have presented the DHHD as a tool for the analysis of motion fields which occur in image and video processing . This technique was applied to three problems in image processing — hurricane tracking, fingerprint reference point identification and identification of critical points in a cardiac optically mapped video.

Intelligent analysis of satellite data is a very important meteorological application. In this context, we developed a method to track the eye of a hurricane over a sequence of satellite image frames. The only data we use is the satellite visual channel images (most other techniques use multiple channels like visual, infrared, *etc*). In this application, motion between two consecutive frames was estimated using a block matching algorithm and then DHHD was used to identify the center of rotation (*i.e.* the hurricane eye). The estimated location of the eye was then used to initialize a level set algorithm which tracked the eye over consecutive video frames. Thus, an automated system for hurricane eye tracking was developed.

In the fingerprint application, the DHHD is used to identify reference points in fingerprints. Reference points are essential for registration of fingerprint images before matching. First the orientation field is extracted from the fingerprint image, after which DHHD is applied to the extracted field for critical point identification (the point with maximum curvature). The technique developed is robust against

noise and against rotation of the fingerprint images.

We also developed a framework for analyzing cardiac optically mapped videos. We have combined techniques developed by cardiologists (using simulated cardiac models) with image processing techniques to develop methods for the analysis of cardiac optical maps. A scheme for studying the morphological structure of a wavefront, and its relation to propagation velocity has been developed in this thesis. Estimation of physical parameters of the cardiac tissue using this model was quite accurate. We also compared two different methods for analyzing cardiac velocity vectors. The performance of the two techniques was evaluated on velocity vectors extracted from cardiac video frames. We demonstrated that the technique proposed in [56] is more effective than the methods given in [57] as the former is able to identify rotational centers in the motion field which the latter is can not predict accurately. Some other methods were also proposed to retrieve useful information from optical mapping data.

Although the algorithms proposed in this thesis show promising results, there are issues which might be improved with further work. A few potential directions of future work are:

- The hurricane tracking algorithm requires periodic initialization using DHHD. At present, the initialization period is specified manually. In future work, a technique for automatically determining the initialization period for a given video sequence could be developed.

- Other channels of data from the satellites, *viz.* visual, infrared, *etc* can be combined with the information from the visual channel to get a more complete hurricane tracking scheme.

- Fingerprints have another ridge structure called delta, where ridges from different directions merge. Detection of the delta structures using the curl free

field should be an interesting area for future study.

- The isopotential detection algorithm using the Canny edge detector might return an edge which is not closed. When the isopotential segment is not closed, the end points of the segment cannot be parametrized using the spline fitting technique. A segmentation method which would ensure that the Canny edge detector returns a closed contour would be another possible direction for further study.

- The analysis of 3D cardiac waves can be performed by parametrizing the surface of a wave by fitting 4^{th} order spline surfaces.

Bibliography

[1] Q. Guo, "Cardiac video analysis using Hodge-Helmholtz decomposition," Master's thesis, University of Alberta, Edmonton, Canada, December 2003.

[2] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, February 1996.

[3] R. Arora, M. K. Das, D. P. Zipes, and J. Wu, "Optical mapping of cardiac arrhythmias," *Indian Pacing and Electrophysiology Journal*, vol. 3, no. 4, pp. 187–196, 2003.

[4] V. G. Fast and A. G. Kleber, "Role of wavefront curvature in propagation of cardiac impulse," *Cardiovascular Research*, vol. 33, no. 2, pp. 258–271, 1997.

[5] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[6] ——, "Determining optical flow: A retrospective." *Artif. Intell.*, vol. 59, no. 1-2, pp. 81–87, 1993.

[7] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[8] M. J. Black and P. Anandan, "The robust estimation of multiple motions: parametric and piecewise-smooth flow fields," *Comput. Vis. Image Underst.*, vol. 63, no. 1, pp. 75–104, 1996.

[9] L. V. Tsap, D. B. Goldgof, and S. Sarkar, "Nonrigid motion analysis based on dynamic refinement of finite element models." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 5, pp. 526–543, 2000.

[10] T. Corpetti, E. Mémin, and P. Pérez, "Dense estimation of fluid flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 365–380, March 2002.

[11] R. P. Wildes, M. J. Amabile, A.-M. Lanzillotto, and T.-S. Leu, "Recovering estimates of fluid flow from image sequence data," *Comput. Vis. Image Underst.*, vol. 80, no. 2, pp. 246–266, 2000.

[12] H. Nogawa, Y. Nakajima, Y. Sato, and S. Tamura, "Acquisition of symbolic description from flow fields: A new approach based on a fluid model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 1, pp. 58–63, 1997.

[13] A. R. Rao and R. C. Jain, "Computerized flow field analysis: Oriented texture fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 7, pp. 693–709, 1992.

[14] T. Corpetti, E. Mémin, and P. Pérez, "Extraction of singular points from dense motion fields: an analytic approach," *Journal of Mathematical Imaging and Vision*, vol. 19, no. 3, pp. 175–198, 2003.

[15] S. Treetasanatavorn, U. Rauschenbach, J. Heuer, and A. Kaup, "Stochastic motion coherency analysis for motion vector field segmentation on compressed video sequences," in *Proc. Int'l Workshop on Image Analysis for Multimedia Interactive Services*, 2005.

[16] P. D. Haggerty, K. Sprietzer, G. Legg, and R. Luczak, "Evolution of nanook, the NOAA/NESDIS near-real-time modis processing system," in *18th International Conference on IIPS*, 2004.

[17] D. Mukherjee and S. Acton, "Cloud tracking by scale space classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 2, pp. 405–415, Feb. 2002.

[18] A. K. Jain and D. Maltoni, *Handbook of Fingerprint Recognition*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003.

[19] A. K. Jain and S. Pankanti, "Automated fingerprint identification and imaging systems," in *Advances in Fingerprint Technology*, H. C. Lee and R. E. Gaensslen, Eds. Elsevier Science, 2001.

[20] A. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 846–859, 2000.

[21] F. H. Fenton, E. M. Cherry, H. M. Hastings, and S. J. Evans, "Multiple mechanisms of spiral wave breakup in a model of cardiac electrical activity," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 12, no. 3, pp. 852–892, 2002.

[22] C. Luo and Y. Rudy, "A model of ventricular cardiac action potential depolarization, repolarization and their interaction," *Circulation Research*, vol. 68, pp. 1501–1526, 1991.

[23] J. Beaumont, N. Davidenko, J. M. Davidenko, and J. Jalife, "Spiral waves in two-dimensional models of ventricular muscle: Formation of a stationary core," *Biophysical Journal*, vol. 52, pp. 1–14, July 1998.

[24] M. W. Kay and R. A. Gray, "Measuring curvature and velocity vector fields for waves of cardiac excitation in 2-D media," *IEEE Trans. on Biomedical Engineering*, vol. 52, no. 1, pp. 50–63, January 2005.

[25] G. E. Morley and D. Vaidya, "Understanding conduction of electrical impulses in the mouse heart using high-resolution video imaging technology," *Microscopy Research and Technique*, vol. 52, pp. 241–250, 2001.

[26] T. Kohlberger, E. Mmin, and C. Schnoerr, "Variational dense motion estimation using the Helmholtz decomposition," in *Scale Space 03*, ser. Lecture notes in computer science, L. Griffin and M. Lillholm, Eds., vol. 2695, June 2003, pp. 432–448.

[27] R. Brad and I. A. Letia, "Cloud motion detection from infrared satellite images," in *Second International Conference on Image and Graphics, SPIE*, vol. 4875, 2002, pp. 408–412.

[28] M. Black, "Recursive nonlinear estimation of discontinuous flow fields," in *Proceedings of the European Conference on Computer Vision*, 1994, pp. 138–145.

[29] T. Corpetti, E. Mémin, and P. Pérez, "Dense fluid flow estimation," IRISA, Tech. Rep. 1352, September 2000.

[30] L. Zhou, C. Kambhamettu, D. Goldgof, K. Palaniappan, and A. Hasler, "Tracking nonrigid motion and structure from 2D satellite cloud images without correspondences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, November 2001.

[31] L. Zhou, "Structure and nonrigid motion analysis from 2D images," PhD in Computer Science, Univ. of Delaware, 2001.

[32] R. M. Ford and R. N. Strickland, "Representing and visualizing fluid flow images and velocimetry data by nonlinear dynamical systems." *CVGIP: Graphical Model and Image Processing*, vol. 57, no. 6, pp. 462–482, 1995.

[33] I. Cohen and I. Herlin, "Non uniform multiresolution method for optical flow and phase portrait models: Environmental applications," *International Journal of Computer Vision*, vol. 33, no. 1, pp. 29–49, 1999.

[34] K. Polthier and E. Preuß, "Identifying vector field singularities using a discrete Hodge decomposition," in *Visualization and Mathematics III*, H.-C. Hege and K. Polthier, Eds. Heidelberg: Springer-Verlag, 2003, pp. 113–134.

[35] Y. Tong, S. Lombeyda, A. N. Hirani, and M. Desbrun, "Discrete multiscale vector field decomposition," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 445–452, 2003.

[36] Q. Guo, M. Mandal, and M. Li, "Efficient Hodge-Helmholtz decomposition of motion fields," *Pattern Recognition Letters*, vol. 26, no. 4, pp. 493–501, March 2005.

[37] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *International Journal on Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[38] R. Malladi, J. Sethian, and B. Vemuri, "Shape modelling with front propagation: A level set approach," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 158–174, 1995.

[39] J. A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry,Fluid Mechanics, Computer Vision and Material Science*. Cambridge University Press, 1996.

[40] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, 1986.

[41] A. K. Jain, L. Hong, and R. M. Bolle, "On-line fingerprint verification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 302–314, 1997.

[42] L. Hong, "Automatic personal identification using fingerprints," Ph.D. dissertation, Michigan State University, 1998.

[43] S. C. Dass, "Markov random field models for directional field and singularity extraction in fingerprint images," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1358–1367, 2004.

[44] K. Chan, Y. Moon, and P. Cheng, "Fast fingerprint verification using subregions of fingerprint images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 95–101, 2004.

[45] L. Hong, Y. Wan, and A. K. Jain, "Fingerprint image enhancement: Algorithm and performance evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777–789, 1998.

[46] A. K. Jain, S. Prabhakar, and L. Hong, "A multichannel approach to fingerprint classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 348–359, 1999.

[47] X. Jiang, M. Liu, and A. Kot, "Reference point detection for fingerprint recognition," in *Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004.*, vol. 1, 2004, pp. 540–543.

[48] B. Palit, A. Basu, and M. K. Mandal, "Application of the discrete Hodge Helmholtz decomposition to image and video processing," To be presented at the First International Conference on Pattern Recognition and Machine Intelligence (PReMI'05), Kolkata, India, during December 18-22, 2005.

[49] D. Tang, Y. Li, J. Wong, S. Po, E. Patterson, W. R. Chen, W. Jackman, and H. Liu, "Characteristics of a charged-coupled-device-based optical mapping system for the study of cardiac arrhythmias," *Journal of Biomedical Optics*, vol. 10, no. 2, March/April 2005.

[50] F. X. Witkowski, L. J. Leon, P. A. Penkoske, R. B. Clark, M. L. Spano, W. L. Ditto, and W. R. Giles, "A method for visualization of ventricular fibrillation: Design of a cooled fiberoptically coupled image intensified ccd data acquisition system incorporating wavelet shrinkage based adaptive filtering," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 8, no. 1, pp. 94–102, March 1998.

[51] D. J. Christini and L. Glass, "Introduction: Mapping and control of complex cardiac arrhythmias," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 12, no. 3, pp. 732–739, 2002.

[52] W. T. Baxter, S. F. Mironov, A. V. Zaitsev, J. Jalife, and A. M. Pertsov, "Visualizing excitation waves inside cardiac muscle using transillumination," *Biophysical Journal*, vol. 80, p. 516530, January 2001.

[53] A. R. S. Kleber and V. Fast, "Role of cell-to-cell coupling, structural discontinuities, and tissue anisotropy in propagation of the electrical impulse," in *Optical Mapping of Cardiac Excitation and Arrhythmias*, D. S. Rosenbaum, Ed. Armonk, NY, Futura, 2001, pp. 137–155.

[54] D. Sung, J. Somayajula-Jagai, P. Cosman, R. Mills, and A. McCulloch, "Phase-shifting prior to spatial filtering enhances optical recordings of cardiac action potential propagation," *Annals of Biomedical Engineering*, vol. 29, no. 10, pp. 854–861, October 2001.

[55] G. K. Rohde, B. M. Dawant, and S. F. Lin, "Correction of motion artifact in cardiac optical mapping using image registration," *IEEE Transactions on Biomedical Engineering*, vol. 52, pp. 338–341, February 2005.

[56] Q. Guo, M. K. Mandal, G. Liu, and K. M. Kavanagh, "Cardiac video analysis using Hodge-Helmholtz field decomposition," *Computers in Biology and Medicine*, In Press: Available online.

[57] T. N. Fitzgerald, D. H. Brooks, and J. Triedman, "Identification of cardiac rhythm features by mathematical analysis of vector fields," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 1, pp. 19–29, Jan 2005.

[58] ——, "Comparative psychometric analysis of vector and isochrone cardiac activation maps," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 5, pp. 847– 855, May 2004.

[59] P. Bayly, B. KenKnight, J. Rogers, R. Hillsley, R. Ideker, and W. Smith, "Estimation of conduction velocity vector fields from epicardial mapping data," *IEEE Transactions on Biomedical Engineering*, vol. 45, no. 5, pp. 563–571, May 1998.

[60] M. S. Spach and J. M. Kootsey, "The nature of electrical propagation in cardiac muscle," *American Jounal of Physiology*, vol. 244(1), pp. H3–22, 1983.

[61] V. S. Zykov and O. L. Morozova, "Speed of spread of excitation in two dimensional excitable medium," *Biophysics*, vol. 24, pp. 739–744, 1979.

[62] T. M. Lillesand and R. W. Kiefer, *Remote Sensing and Image Interpretation*, 3rd ed. Wiley, 1994.

[63] P. Gamba, "Meteorological structures shape description and tracking by means of BI-RME matching," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 1151–1161, 1999.

[64] K. Palaniappan, C. Kambhamettu, A. Hasler, and D. Goldgof, "Structure and semi-fluid motion analysis of stereoscopic satellite images for cloud tracking," in *Proceedings of the Fifth International Conference on Computer Vision*, 1995, pp. 659–665.

[65] F. Barbaresco and B. Monnier, "Rain clouds tracking with radar image processing based on morphological skeleton matching," in *Proceedings of the International Conference on Image Processing*, vol. 1, 2001, pp. 830–833.

[66] T. Bretschneider and L. Yikun, "On the problems of locally defined content vectors in image databases for large images," in *Proceedings of the International Conference on Information Communications and Signal Processing IEEE PCM*, vol. 3, 2003, pp. 1604–1608.

[67] C. Papin, P. Bouthemy, E. Mémin, and G. Rochard, "Tracking and characterization of highly deformable cloud structures," in *Proceedings of the 6th European Conference on Computer Vision-Part II*. Springer-Verlag, 2000, pp. 428–442.

[68] C. Papin, P. Bouthemy, and G. Rochard, "Unsupervised segmentation of low clouds from infrared meteosat images based on a contextual spatio-temporal labeling approach," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 40, no. 1, pp. 104–114, January 2002.

[69] J. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of Visual Communication and Image Representation*, vol. 6, no. 4, pp. 348–365, December 1995.

[70] J.-M. Odobez, P. Bouthemy, and F. Spindler, "Motion2D," http://www.irisa.fr/Vista/Motion2D/.

[71] A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnorr, "Discontinuity-preserving computation of variational optic flow in real-time," in *Proc. 5th International Conference Scale-Space*. Springer, Berlin, 2005, pp. 279–290.

[72] A. Bazen and S. Gerez, "Systematic methods for the computation of the directional fields and singular points of fingerprints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 905–919, 2002.

[73] A. Jain, L. Hong, S. Pankanti, and R. Bolle, "An identity authentication system using fingerprints," *Proceedings of the IEEE*, vol. 85, no. 9, September 1997.

[74] J. Zhou and J. Gu, "A model-based method for the computation of fingerprints' orientation field," *IEEE Trans. on Image Processing*, vol. 13, no. 6, June 2004.

[75] "Image processing toolbox for use with MATLAB, Users Guide version 5, The Mathworks, Inc., Natick, MA," 2005.

[76] R. C. Gonzalez and R. E. Woods, *Digital Image processing*. Reading, Mass., Addison-Wesley, 1992.

[77] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, August 1979.

[78] P. Lancaster and K. Salkauskas, *Curve and surface fitting: An introduction*. Academic, London, UK, 1986.

[79] C. de Boor, *Spline Toolbox For Use with MATLAB, Users Guide Version 3*. The MathWorks, Inc., Natick, MA, 2004.

[80] E. Kreyszig, *Advanced Engineering Mathematics*, 4th ed. New York, NY, USA: John Wiley & Sons, 1979.

Appendix A

Code for DHHD

The DHHD algorithm is a central theme of this thesis. Here we provide an implementation of the algorithm in Matlab. The given algorithm takes as input parameter, a vector field and decomposes it into three components.

```
function [curl_free,div_free,har_rem]=dhhd(mv)
%% Rearrange the input vector field into its components
mv_x=mv(:,:,1); %x-Coordinates of grid
mv_y=mv(:,:,2); %y-Coordinates of grid
mv_u=mv(:,:,3); %x-component of vector
mv_v=mv(:,:,4); %y-component of vector

%% Plot original motion field
figure;
quiver(mv_x,mv_y,mv_u,mv_v);
axis tight
title('Original Motion Field')
%% generate the curl-free and divergence-free curl_free_potntial
%% functions
[curl_free_pot,div_free_pot]=potential_calculation...
(mv_x,mv_y,mv_u,mv_v);
%% display the curl-free potential function E
figure;
meshc(mv_x,mv_y,curl_free_pot);
title('curl free potential function');
%% display the divergence-free potential function W
figure;
meshc(mv_x,mv_y,div_free_pot);
title('divergence free potential function');
%% generate the curl-free field
[uu_E,vv_E]=gradient(curl_free_pot);
%% display the curl-free field
figure;
quiver(mv_x,mv_y,uu_E,vv_E);
axis tight
title('Curl free component of the field');
%% Store curl free field
curl_free(:,:,1) = uu_E;
curl_free(:,:,2) = vv_E;
```

```
%% CODE FOR COLOURED PLOTS
figure;
surf(mv_x,mv_y,div_free_pot,'FaceColor','interp',...
    'EdgeColor','none',...
    'FaceLighting','phong')
axis tight
view(-50,30)
title('Divergence free potential');
%camlight left
hold_state = ishold;
hold on;
a = get(gca,'zlim');
zpos = a(1); % Always put contour below the plot.
%% Get D contour data
[cc,hh] = contour3(mv_x,mv_y,div_free_pot,7);
for i = 1:length(hh)
    zz = get(hh(i),'Zdata');
    set(hh(i),'Zdata',zpos*ones(size(zz)));
end
figure;
surf(mv_x,mv_y,curl_free_pot,'FaceColor','interp',...
    'EdgeColor','none',...
    'FaceLighting','phong')

axis tight
title('curl free potential')
view(-50,30)
camlight left
hold_state = ishold;
hold on;
a = get(gca,'zlim');
zpos = a(1); % Always put contour below the plot.
%% Get D contour data
[cc,hh] = contour3(mv_x,mv_y,curl_free_pot,7);
for i = 1:length(hh)
    zz = get(hh(i),'Zdata');
    set(hh(i),'Zdata',zpos*ones(size(zz)));
end

%% generate the divergence-free field
[tmp_u,tmp_v]=gradient(div_free_pot);
uu_W=tmp_v; vv_W=-tmp_u;
%% display the divergence-free field
figure;
quiver(mv_x,mv_y,uu_W,vv_W);% axis ij; axis([-box,box,-box,box]);
title('Divergence free component of the field');
axis tight
%% Store the divergence free field
div_free(:,:,1) = uu_W;
div_free(:,:,2) = vv_W;

%% calculate the magnitude field of the input motion field
mag0=sqrt(mv_u.^2+mv_v.^2);
```

```
%% find the largest magnitude of the input motion field
max_mag0=max(max(mag0));
%% add the curl-free field and the divergence-free field
 U1=uu_E+uu_W; V1=vv_E+vv_W;

%% calculate the magnitude field of the added motion field
mag1=sqrt(U1.^2+V1.^2);
%% find the largest magnitude of the added motion field
max_mag1=max(max(mag1));
%% calculate the scaling factor, r in the thesis
ratio=max_mag0/max_mag1;
%% solve for the harmonic remainder
ru=mv_u-ratio*U1; rv=mv_v-ratio*V1;
%% display the harmonic remainder
figure;
quiver(mv_x,mv_y,ru,rv);
axis tight
title('Harmonic remainder');
%% Store harmonice remainder
har_rem(:,:,1) = ru;
har_rem(:,:,2) = rv;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [potE,potW]=potential_calculation(mv_x,mv_y,mv_u,mv_v)
%% potE: the curl-free potential function W
%% potW: the divergence-free potential funciton W
%% mv_x: horizontal coordinates of the grids
%% mv_y: vertical coordinates of the grids
%% mv_u: horizontal components of the input motion field
%% mv_v: vertical components of the input motion field
[M,N]=size(mv_x); S1=construct_S1(mv_x,mv_y); %
%% construct the element matrix S1
[sa,sb]=size(S1);
Sr=S1(2:sa,2:sb);
ISr=inv(Sr);
potE=zeros(M,N);
potW=zeros(M,N);
[B,C]=construct_BC(mv_x,mv_y,mv_u,mv_v);
%% construct the two vector B and C
Br=B(2:sa); Cr=C(2:sa); Er=ISr*Br;
%% solve for the (L-1)x1 vector Er
Wr=ISr*Cr;
%% solve for the (L-1)x1 vector Wr
E_pot=[0;Er];
%% reconstruct the Lx1 vector E
W_pot=[0;Wr];
%% reconstruct the Lx1 vector W
potE=reshape(E_pot,M,N);
%% re-organize the MxN potential surface E
potW=reshape(W_pot,M,N);
%% re-organize the MxN potential surface W

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function S1=construct_S1(mv_x,mv_y)
%% mv_x: horizontal coordinates of the grids
%% mv_y: vertical coordinates of the grids
%% S1: the element matrix S1
%% generate the Node Table
mesh=mesh_points(mv_x,mv_y);
%% generate the Grid Table
triangle=triangle_definition(mv_x,mv_y);

%% generate the Basis Gradient Table
gradphi=phi_gradient(triangle,mesh);
M=size(mesh,1); S1=zeros(M,M);
%% calculate the element matrix S1
for i=1:M
    %% search for the neighboring triangles and neighboring
    %% nodes of a reference node
    [neighbor_triangle neighbor_point]=get_neighbor(i,triangle);
    L=length(neighbor_triangle);
    for k=1:L
        t=neighbor_triangle(k);
        tri=triangle(t,:);
        order_in_triangle=find(tri==i);
        del_phi01=gradphi(t,order_in_triangle,1);
        del_phi02=gradphi(t,order_in_triangle,2);
        for j=1:3 mm=tri(j);
            del_phi11=gradphi(t,j,1);
            del_phi12=gradphi(t,j,2);
            S1(i,mm)=S1(i,mm)+del_phi01*del_phi11 + ...
             del_phi02*del_phi12;
        end;
    end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [B,C]=construct_BC(mv_x,mv_y,mv_u,mv_v)
%% mv_x: horizontal coordinates of the grids
%% mv_y: vertical coordinates of the grids
%% mv_u: horizontal component of the input motion field
%% mv_v: vertical component of the input motion field
%% B: vector B
%% C: vector C
%% generate the Node Table
mesh=mesh_points(mv_x,mv_y); %% generate the Grid Table
triangle=triangle_definition(mv_x,mv_y);

%% generate the Average Vector Table
%changed mv_x(1,2)                    mv_y(2,1)
delta_x=mv_x(1,2)-mv_x(1,1); delta_y=mv_y(2,1)-mv_y(1,1);
UV=triangle_uv(triangle,mesh,mv_x(1,1),mv_y(1,1),..
delta_x,delta_y,mv_u,mv_v);
%% generate the Basis Gradient Table
gradphi=phi_gradient(triangle,mesh);
```

```
M=size(mesh,1); B=zeros(M,1); C=zeros(M,1);
%% calculate the vectors B and C
for i=1:M
    %% search for the neighboring triangles and neighboring nodes
    %% of a reference node i
    [neighbor_triangle neighbor_point]=get_neighbor(i,triangle);
    L=length(neighbor_triangle);
    for k=1:L
        t=neighbor_triangle(k);
        tri=triangle(t,:);
        order_in_triangle=find(tri==i);
        del_phi01=gradphi(t,order_in_triangle,1);
        del_phi02=gradphi(t,order_in_triangle,2);
        B(i)=B(i)+del_phi01*UV(t,1)+del_phi02*UV(t,2);
        C(i)=C(i)-del_phi01*UV(t,2)+del_phi02*UV(t,1);
    end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function xy=mesh_points(mv_x,mv_y)
%% generate the Node Table
%% mv_x: horizontal coordinates of the grids
%% mv_y: vertical coordinates of the grids
%% xy: the coordinates of all nodes
sz=size(mv_x);
MN=prod(sz);
xx=reshape(mv_x,MN,1);
yy=reshape(mv_y,MN,1);
xy=zeros(MN,2);

for i=1:MN
    xy(i,1)=mv_y(i);
    xy(i,2)=mv_x(i);
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function triangles=triangle_definition(mv_x,mv_y)
%% generate the Grid Table, which defines the grid topology
%% mv_x: horizontal coordinates of the grids
%% mv_y: vertical coordinates of the grids
%% triangles: Nodes of all triangular meshes,
%% i.e., the grid topology
[M,N]=size(mv_x);
period=(M-1);
total_tri=2*period*(N-1);
triangles=zeros(total_tri,3);
for i=1:N-1
    for j=1:period
        k=(i-1)*period+j;
        start=floor((k-1)/period)*M+mod(k-1,period)+1;
        triangles(2*k-1,:)=[start,start+M+1,start+M];
        triangles(2*k,:)=[start,start+1,start+M+1];
```

```
    end;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function UV = ...
triangle_uv(triangle,mesh,start_x,start_y,delta_x,delta_y,uu,vv)
%% generate the Average Vector Table
%% triangle: the Grid Table
%% mesh: the Node Table
%% start_x: start coordinate in horizontal direction
%% start_y: start coordinate in vertical direction
%% delta_x: step in horizontal direction
%% delta_y: step in vertical direction
%% uu: horizontal component of the input motion field
%% vv: vertical component of the input motion field
%% UV: The resulted Average Vector Table
M=size(triangle,1); UV=zeros(M,2);

for i=1:M
    loc1=mesh(triangle(i,1),:);
    loc2=mesh(triangle(i,2),:);
    loc3=mesh(triangle(i,3),:);
    ind_y1=floor((loc1(1)-start_y)/delta_y)+1;
    ind_x1=floor((loc1(2)-start_x)/delta_x)+1;
    ind_y2=floor((loc2(1)-start_y)/delta_y)+1;
    ind_x2=floor((loc2(2)-start_x)/delta_x)+1;
    ind_y3=floor((loc3(1)-start_y)/delta_y)+1;
    ind_x3=floor((loc3(2)-start_x)/delta_x)+1;
    UV(i,1)=(uu(ind_y1,ind_x1)+uu(ind_y2,ind_x2)...
     +uu(ind_y3,ind_x3))/3;
    UV(i,2)=(vv(ind_y1,ind_x1)+vv(ind_y2,ind_x2)...
     +vv(ind_y3,ind_x3))/3;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function gradphi=phi_gradient(triangle,mesh)
%% generate the Basis Gradient Table
%% triangle: the Grid Table
%% mesh: the Node Table
%% gradphi: the resulted Basis Gradient Table
M=size(triangle,1);
gradphi=zeros(M,3,2);
area2=get_area2(triangle(1,:),mesh);
%% calculate the 2*triangle_area
AB=zeros(3,2);
for i=1:M y0=mesh(triangle(i,1),1);
    x0=mesh(triangle(i,1),2);
    y1=mesh(triangle(i,2),1);
    x1=mesh(triangle(i,2),2);
    y2=mesh(triangle(i,3),1);
    x2=mesh(triangle(i,3),2);
    AB(1,1)=y1-y2;
```

```
    AB(1,2)=x2-x1;
    AB(2,1)=y2-y0;
    AB(2,2)=x0-x2;
    AB(3,1)=y0-y1;
    AB(3,2)=x1-x0;
    for j=1:3
        for k=1:2
            gradphi(i,j,k)=-AB(j,k);
        end;
    end;
end; gradphi=gradphi/area2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [neighbor_triangle,neighbor_point]= ...
get_neighbor(node,triangle)
%% get the neighboring triangles and neighboring nodes of a node
%% node: the reference node
%% triangle: the Grid Table
%% neighbor_triangle: the neighbor triangles of the reference node
%% neighbor_point: the neighbor points of the reference node
M=size(triangle,1);
neighbor_triangle=[];
for i=1:M
    temp=triangle(i,:);
    if find(temp==node)>0
        neighbor_triangle=[neighbor_triangle i];
    end;
end;
neighbor_point=[];
N=length(neighbor_triangle);
for i=1:N
    temp=triangle(neighbor_triangle(i),:);
    for j=1:3
        if find(neighbor_point==temp(j))>0
            continue;
        else
            neighbor_point=[neighbor_point temp(j)];
        end;
    end;
end;
neighbor_point=sort(neighbor_point);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function area2=get_area2(triangle,mesh)
%% calculate the 2*triangle_area of a triangle
%% triangle: the Grid Table
%% mesh: the Node Table
%% area2: the resulted 2*triangle_area
x0=mesh(triangle(1),1);
y0=mesh(triangle(1),2);
x1=mesh(triangle(2),1);
y1=mesh(triangle(2),2);
```

```
x2=mesh(triangle(3),1);
y2=mesh(triangle(3),2);
area2=abs((x1-x0)*(y2-y0)-(x2-x0)*(y1-y0));
```